WebObjects Technical Overview

Session 701

WebObjects Technical Overview

Steve Hayman
Consulting Engineer, Apple Education
Toronto

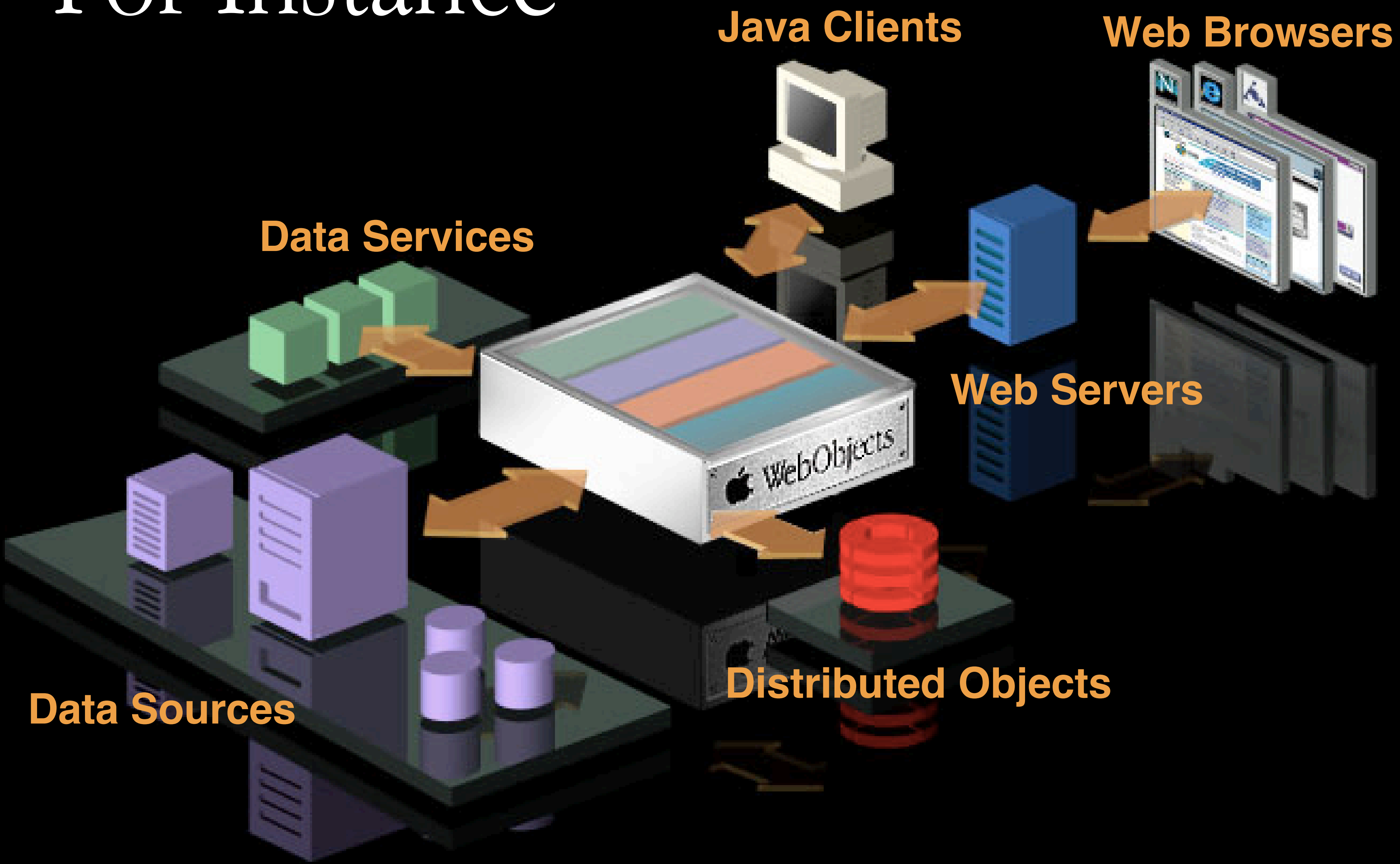
What You’ll Learn

- WO Tools
- WO Frameworks
- WO Deployment
- What’s the deal with WO 5.1?

What WebObjects Does

- It is an Application Server
- Who invented that term anyway?
- It is more than just that
    - Developer Tools
    - Pre-written objects (Frameworks) ← **The best part!**
    - Runtime environment
    - Monitoring and control tools

For Instance



Who WebObjects Is For

- Developers
- Very powerful
- Steep learning curve
- But that is OK

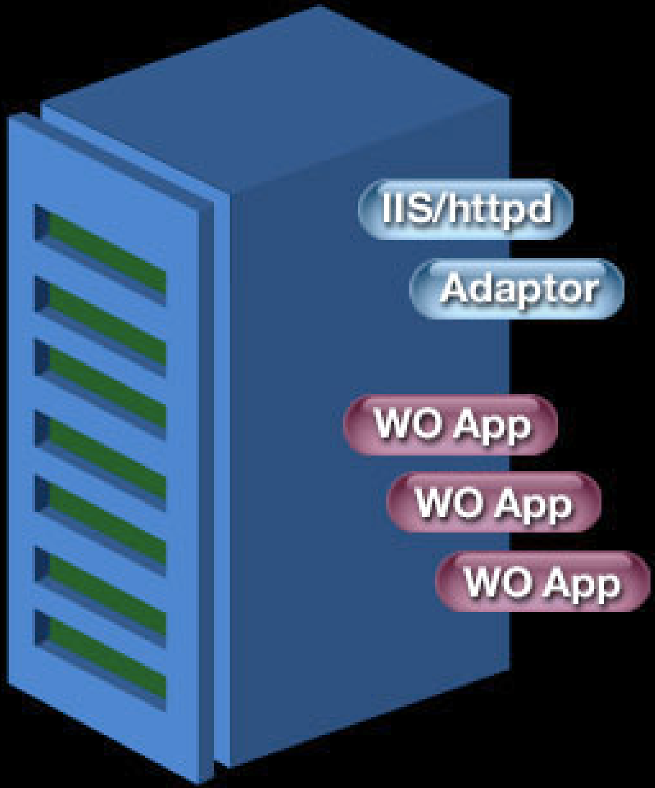
WebObjects Works With . . .

- Java
- Databases—JDBC ,Oracle, SQL Server, OpenBase, . . .
- JNDI Servers—LDAP . . .
- Different UIs—HTML, WAP, PDF, SVG, SMIL, XML
- Most web servers—Apache, Netscape, IIS
- Former competitors—WebLogic . . .
- Lots of third-party objects and APIs
- Open open open open open open open open

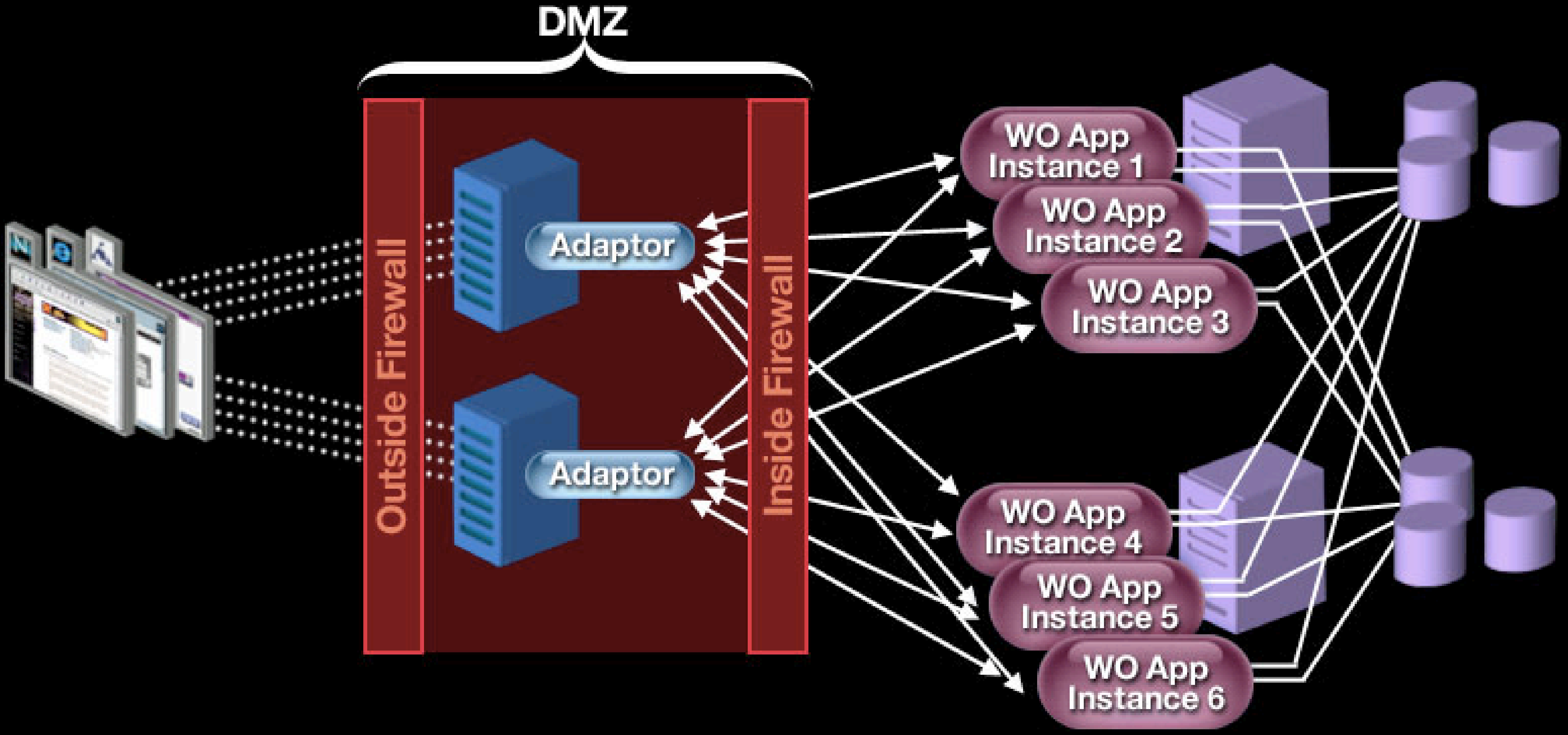
Where You Can Use WebObjects 5

- Develop on Mac OS X or Windows 2000
- Deploy on Mac OS X Server, Windows 2000, Solaris
    - (Theoretically, anywhere with Java 2)
    - And now—Servlet Deployment
- Mix and match as needed
- Start small and grow bigger

Start With This . . .



. . And Scale Easily to This



Browsers

Multiple HTTP Servers

Multiple WO App Servers

Multiple Databases

Key Ideas

- Web Components (front-end)
- Enterprise Objects (back-end)
- Kept completely separate
- Bound together at runtime

Web Components

HTML templates

+

<WEBOBJECT> tags that indicate where something interesting and dynamic happens

+

Bindings that fill in the blanks

+

Lots of pre-written objects to re-use

WO Components

More about this later but this is what they look like…

Main.wo/

- Main.html
- Main.wod

```
<HTML>
This product
<WEBOBJECT NAME=Image4>
costs
<WEBOBJECT NAME=String1>
<HR>
<WEBOBJECT NAME=Link1>
   Click here
</WEBOBJECT>   to order.
```

```
Image4: WOImage {
   src = “pic.gif”; }
String1: WOString {
  value = product.cost
  numberFormat =
         “$#,###”;}
Link1: WOHyperlink {
   action = placeOrder;  }
```

Main.java

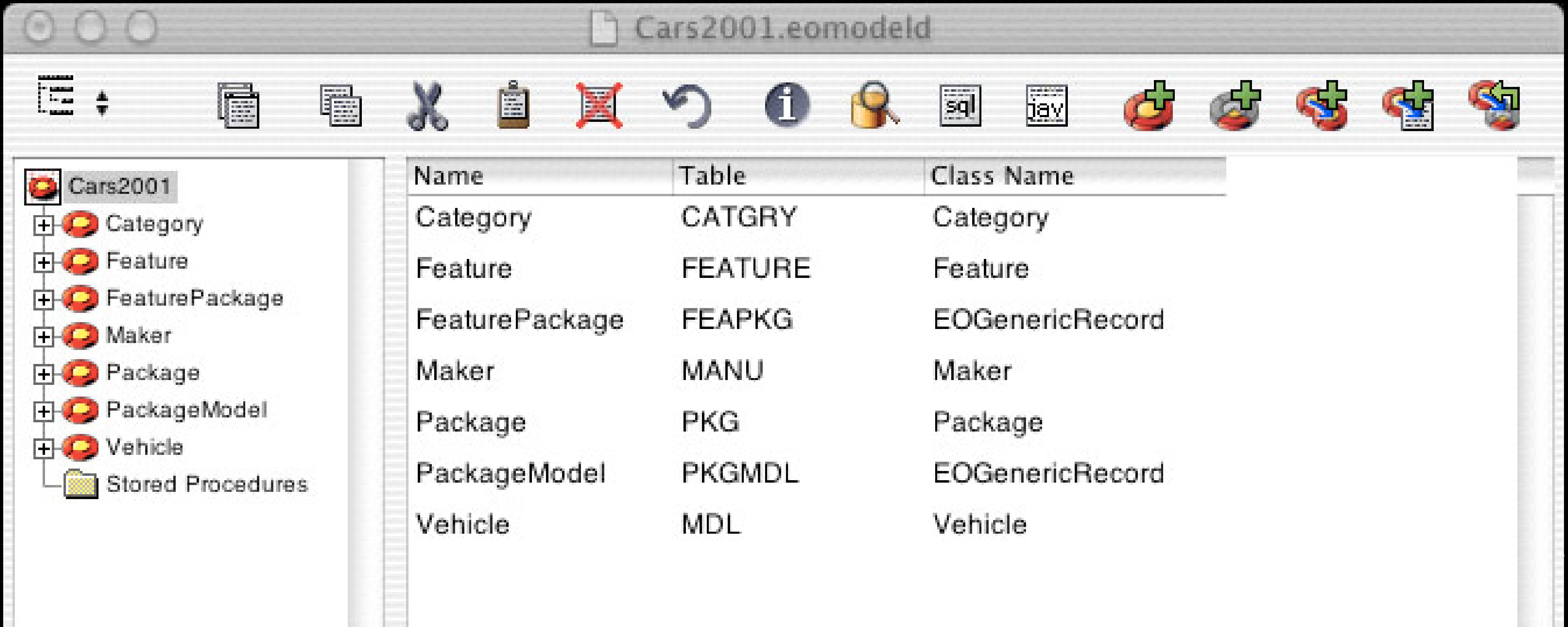
```
public class Order extends WOComponent {
   private ShoppingItem product;
   public WOComponent placeOrder( ) {
     session().addToShoppingCart(product);
     return pageWithName(“Checkout”);
   }
}
```

What Comes Out of a Component?

- A string—of your choice
  - Plain text
  - HTML
  - Image data
  - JavaScript commands
  - AppleScript commands
  - PDF, SMIL, SVG, XML, ...

Enterprise Objects

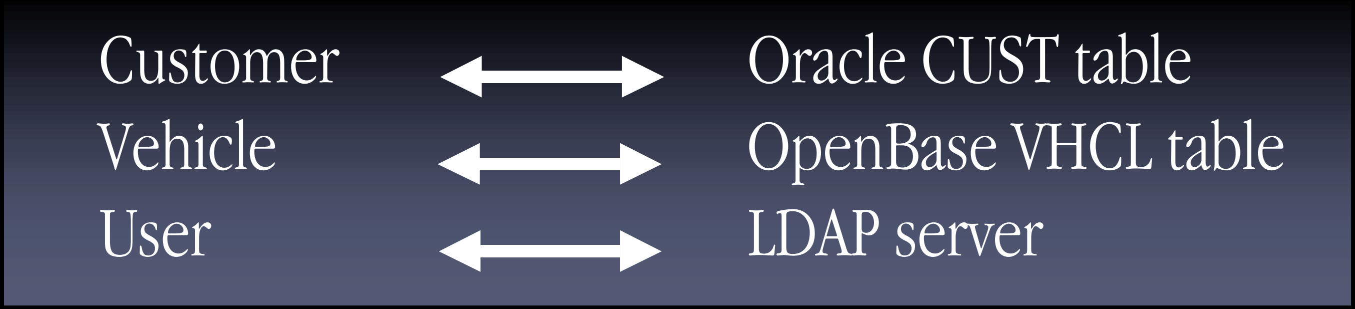
- The back-end database connection technology
- Driven by a Model



Name	Table	Class Name
Category	CATGRY	Category
Feature	FEATURE	Feature
FeaturePackage	FEAPKG	EOGenericRecord
Maker	MANU	Maker
Package	PKG	Package
PackageModel	PKGMDL	EOGenericRecord
Vehicle	MDL	Vehicle

Enterprise Objects

- Java objects that implement your business rules . . .
    - Customer, Order, Product, Vehicle, Package, Option
- JDBC/JNDI Adaptors that speak to databases
    - Oracle, OpenBase, LDAP . . .
- Models that connect objects to databases



Customer ↔ Oracle CUST table

Vehicle ↔ OpenBase VHCL table

User ↔ LDAP server

- Relationships between objects
    - Customer has a list of Vehicles

EOF Is the Real Gem Here!

- Pure business objects
- Totally UI independent
- Totally database independent
- Persistence is completely managed for you

. . All Bound Together . . .

- WO app
  - Loads components
  - Finds **<WEBOBJECT>** tags
  - Fetches and sends messages to Enterprise Objects
  - Mixes results into the component
- User sees plain HTML (PDF, WAP, XML)
- Customize the entire process at runtime if needed

. . .With Advanced Session Management

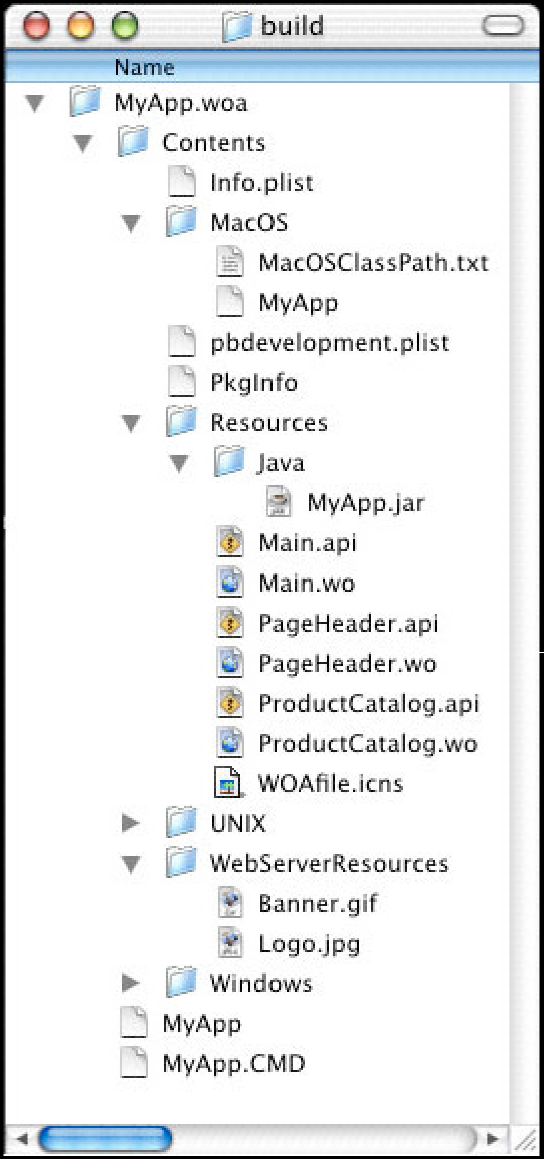
- WO manages an extensible Session object for each individual user
- Also an application object shared by all
- Complete control over session creation and archiving
- Extend Session to add security, shopping carts, user preferences, personalization and so on

. . .And Runtime Monitoring Tools

- Monitor
    - WO application that manages your big honkin’ deployment environment
- Or—Servlet Deployment
    - Just copy your app to the right place for WebLogic, Tomcat, etc

What Is It You’re Creating Anyway?

You Build This . . .

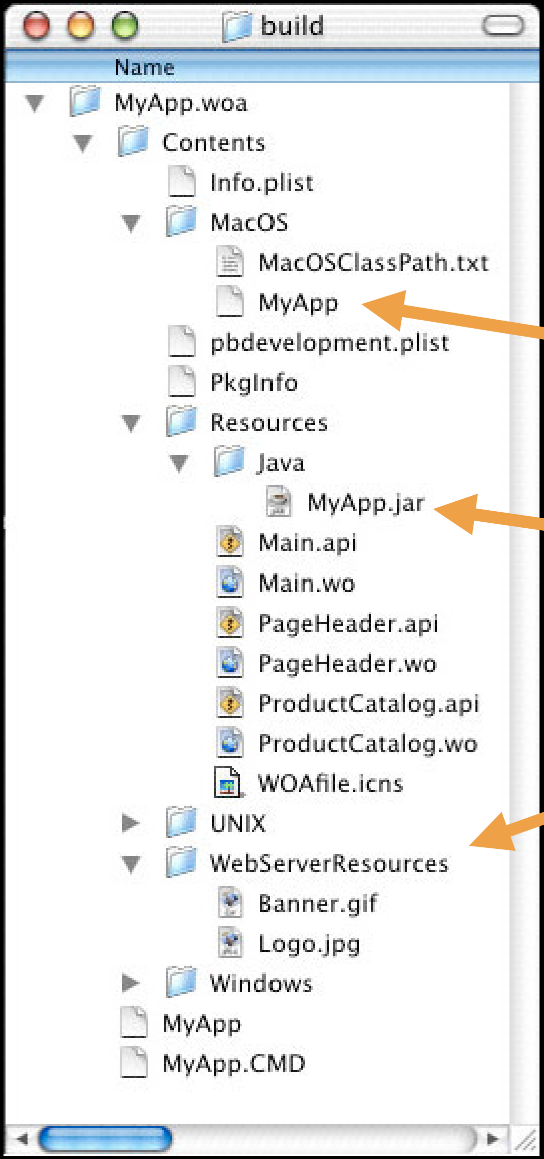


```
MyApp.woa
  Contents
    Info.plist
    MacOS
      MacOSClassPath.txt
      MyApp
    pbdevelopment.plist
    PkgInfo
    Resources
      Java
        MyApp.jar
      Main.api
      Main.wo
      PageHeader.api
      PageHeader.wo
      ProductCatalog.api
      ProductCatalog.wo
      WOAfile.icns
    UNIX
    WebServerResources
      Banner.gif
      Logo.jpg
    Windows
  MyApp
  MyApp.CMD
```

A bunch of WO components
+ Your own Java code
+ Images and other resources
+ References to frameworks

A big honkin' **MyApp.woa** folder

You Build This . . .



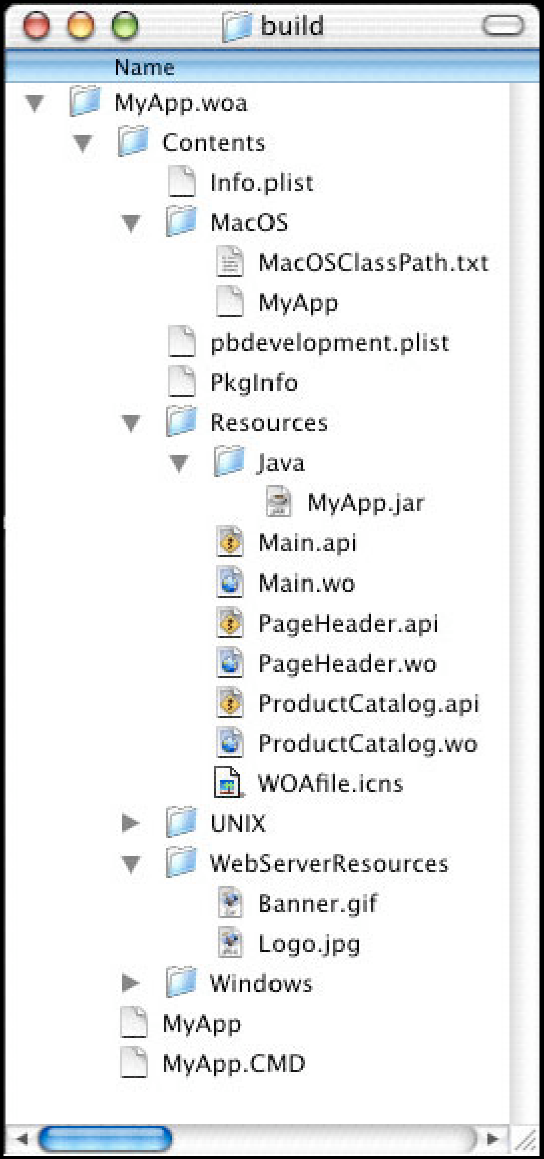
build

Name

```
▼ MyApp.woa
    ▼ Contents
        Info.plist
        ▼ MacOS
            MacOSClassPath.txt
            MyApp
        pbdevelopment.plist
        PkgInfo
        ▼ Resources
            ▼ Java
                MyApp.jar
            Main.api
            Main.wo
            PageHeader.api
            PageHeader.wo
            ProductCatalog.api
            ProductCatalog.wo
            WOAfile.icns
        ▶ UNIX
        ▼ WebServerResources
            Banner.gif
            Logo.jpg
        ▶ Windows
    MyApp
    MyApp.CMD
```

Shell script
runs Java with this jar file
and arranges to find images here.

Launch It From Project Builder



```
▼ MyApp.woa
    ▼ Contents
        Info.plist
        ▼ MacOS
            MacOSClassPath.txt
            MyApp
        pbdevelopment.plist
        PkgInfo
        ▼ Resources
            ▼ Java
                MyApp.jar
            Main.api
            Main.wo
            PageHeader.api
            PageHeader.wo
            ProductCatalog.api
            ProductCatalog.wo
            WOAfile.icns
        ▶ UNIX
        ▼ WebServerResources
            Banner.gif
            Logo.jpg
        ▶ Windows
    MyApp
    MyApp.CMD
```

PB executes the MyApp script

java -classpath SomeHumongousPath Application

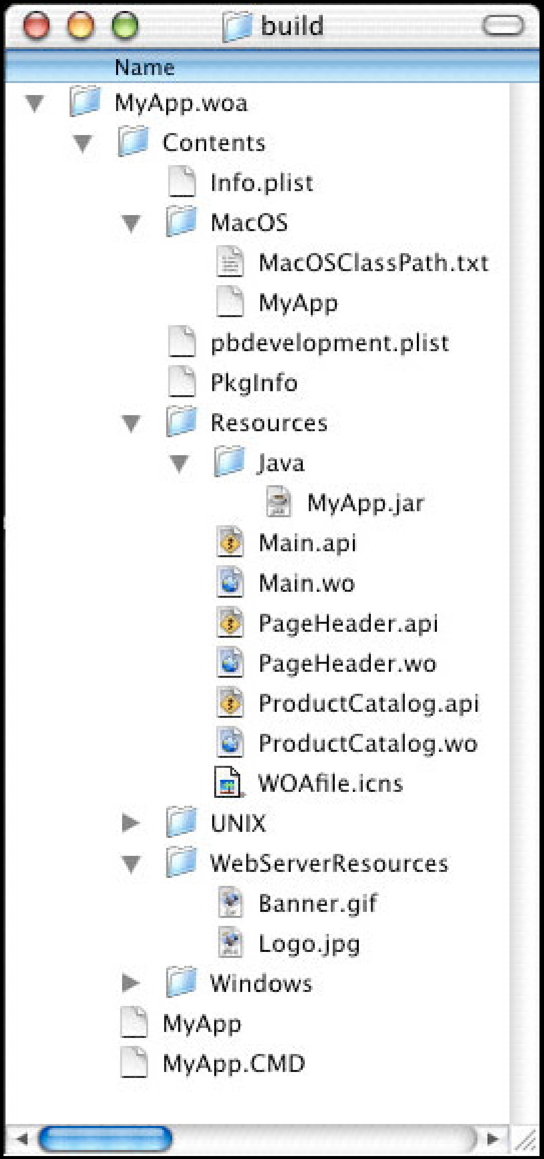
MyApp picks a port number and figures out its own URL

http://localhost:31416/cgi-bin/WebObjects/MyApp

Your browser opens the URL.

Direct Connect

Direct Connect



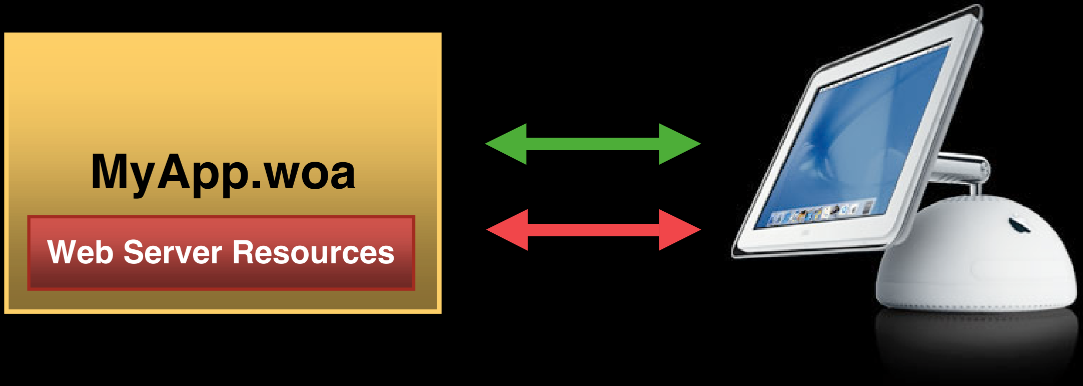
build

Name

- MyApp.woa
  - Contents
    - Info.plist
    - MacOS
      - MacOSClassPath.txt
      - MyApp
    - pbdevelopment.plist
    - PkgInfo
    - Resources
      - Java
        - MyApp.jar
      - Main.api
      - Main.wo
      - PageHeader.api
      - PageHeader.wo
      - ProductCatalog.api
      - ProductCatalog.wo
      - WOAfile.icns
    - UNIX
    - WebServerResources
      - Banner.gif
      - Logo.jpg
    - Windows
  - MyApp
  - MyApp.CMD

🟥 ↔ **Requests for Images**

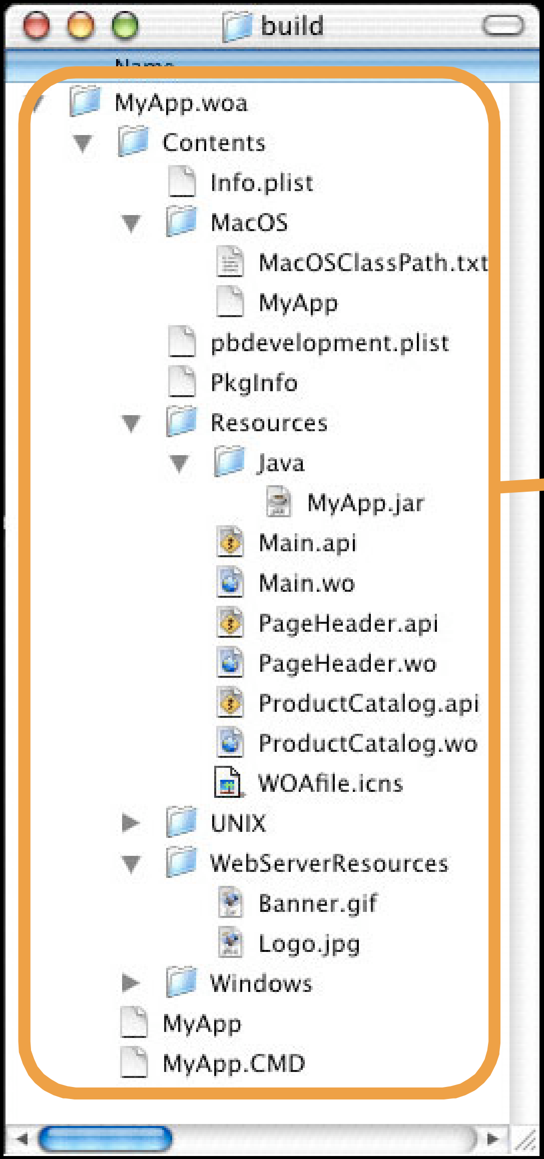
🟩 ↔ **Requests for HTML**



- No web server involved at all
- Browser connects directly to WO app
- Useful during development but not for deployment

When You’re Ready to Deploy

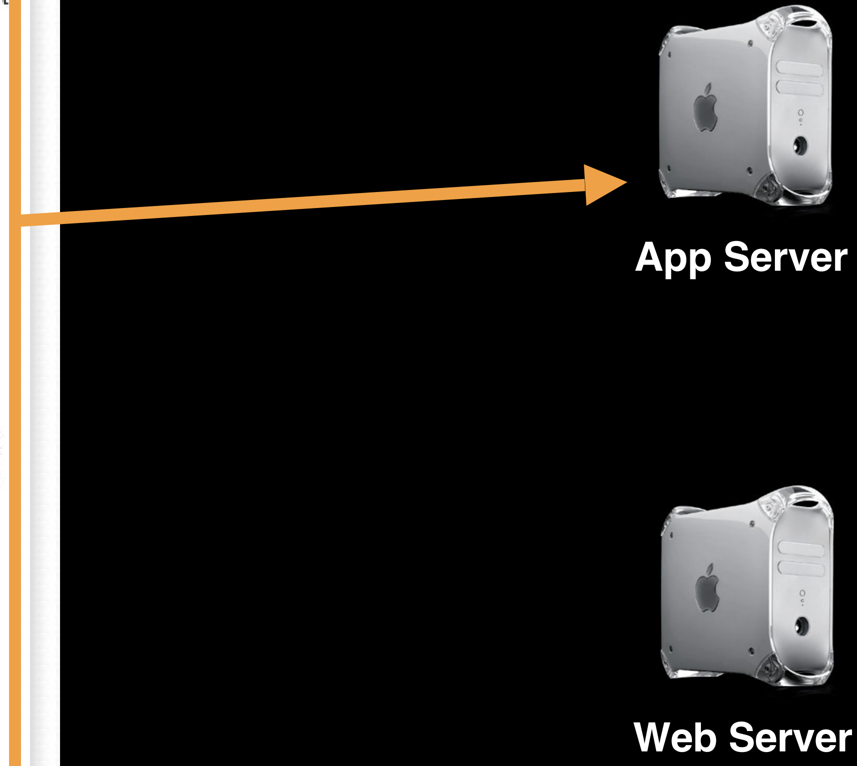
Simple Install



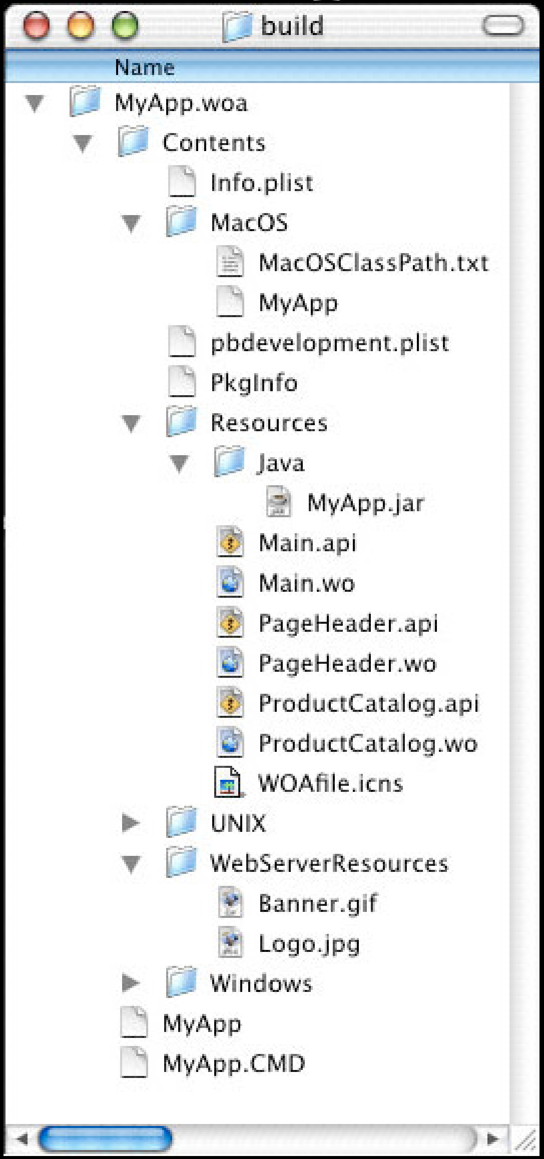
build

Name

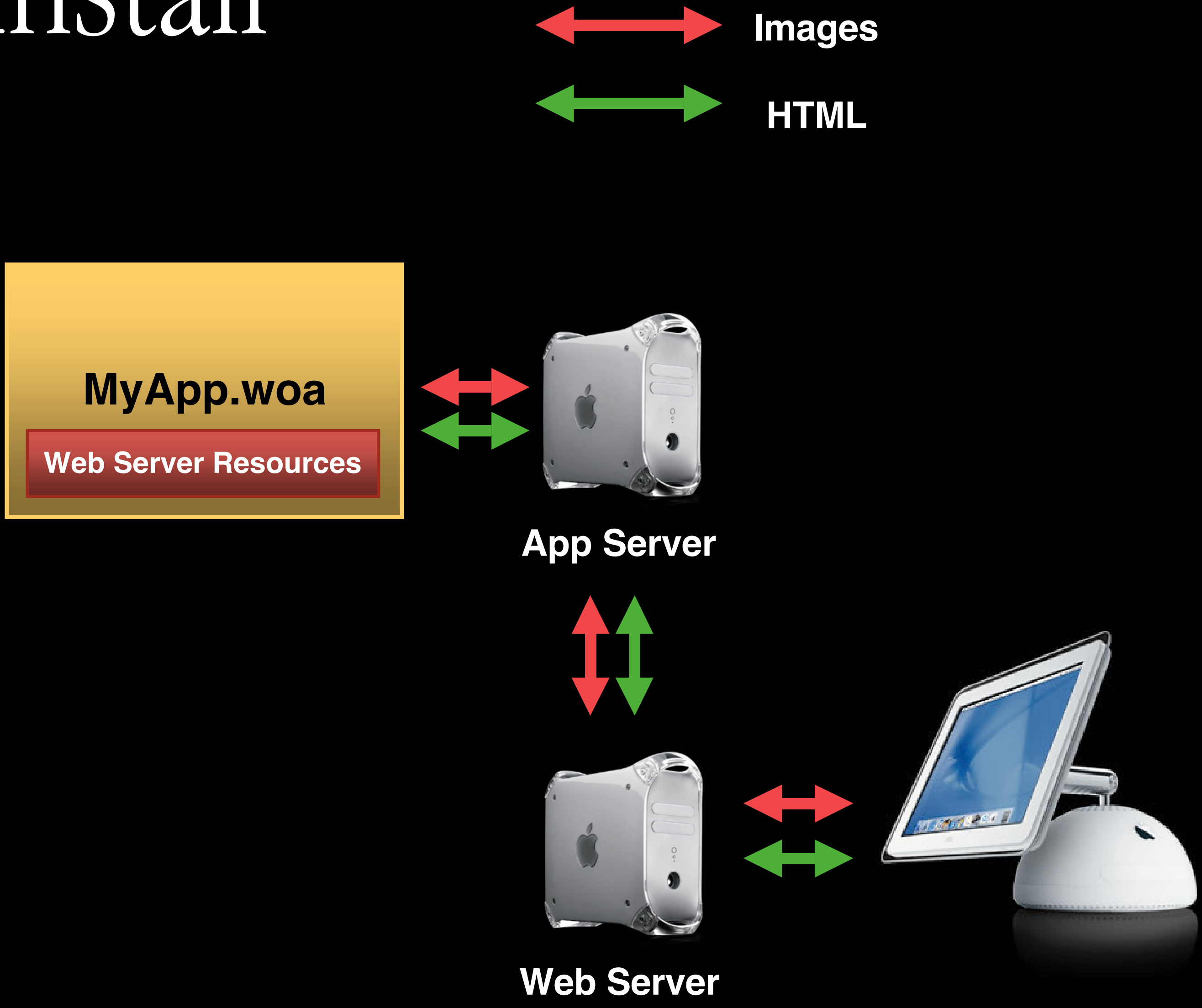
```
MyApp.woa
  Contents
    Info.plist
    MacOS
      MacOSClassPath.txt
      MyApp
    pbdevelopment.plist
    PkgInfo
    Resources
      Java
        MyApp.jar
      Main.api
      Main.wo
      PageHeader.api
      PageHeader.wo
      ProductCatalog.api
      ProductCatalog.wo
      WOAfile.icns
    UNIX
    WebServerResources
      Banner.gif
      Logo.jpg
    Windows
  MyApp
  MyApp.CMD
```



Simple Install

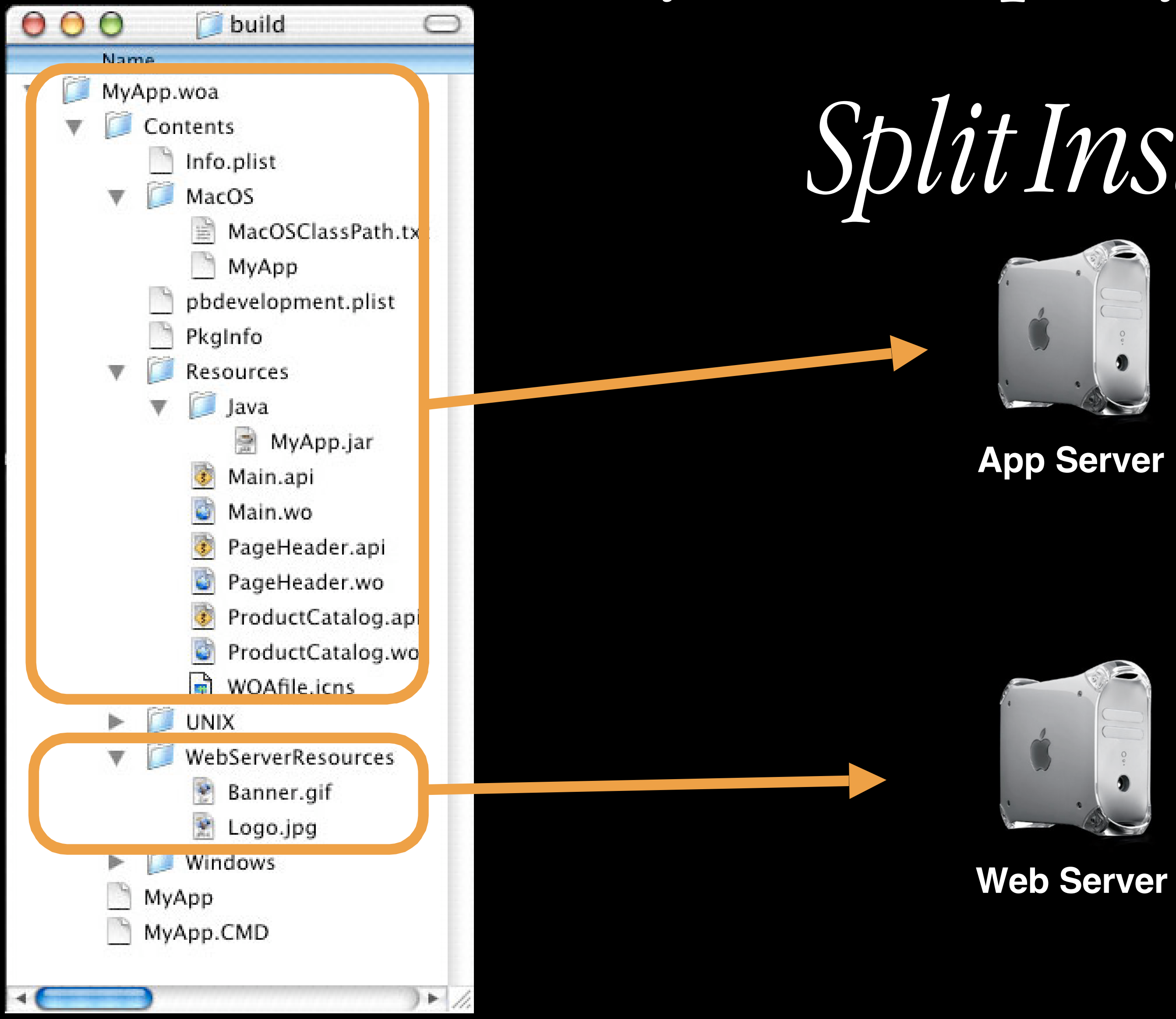


- MyApp.woa
  - Contents
    - Info.plist
    - MacOS
      - MacOSClassPath.txt
      - MyApp
    - pbdevelopment.plist
    - PkgInfo
    - Resources
      - Java
        - MyApp.jar
      - Main.api
      - Main.wo
      - PageHeader.api
      - PageHeader.wo
      - ProductCatalog.api
      - ProductCatalog.wo
      - WOAfile.icns
    - UNIX
    - WebServerResources
      - Banner.gif
      - Logo.jpg
    - Windows
  - MyApp
  - MyApp.CMD



A Better Way to Deploy

Split Install

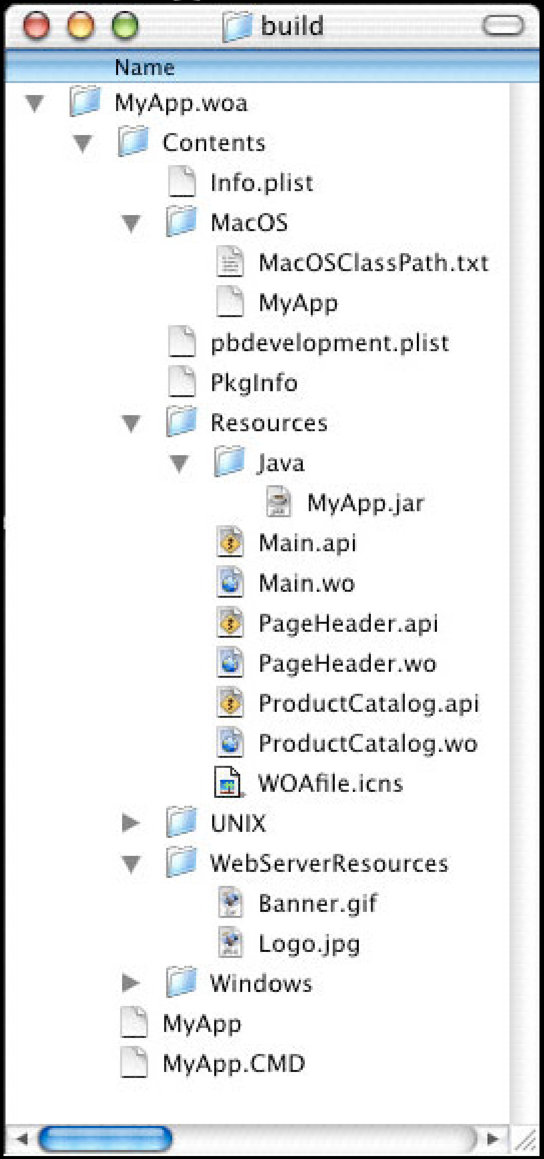


build

Name

```
MyApp.woa
  Contents
    Info.plist
    MacOS
      MacOSClassPath.tx
      MyApp
    pbdevelopment.plist
    PkgInfo
    Resources
      Java
        MyApp.jar
      Main.api
      Main.wo
      PageHeader.api
      PageHeader.wo
      ProductCatalog.api
      ProductCatalog.wo
      WOAfile.icns
    UNIX
    WebServerResources
      Banner.gif
      Logo.jpg
    Windows
  MyApp
  MyApp.CMD
```

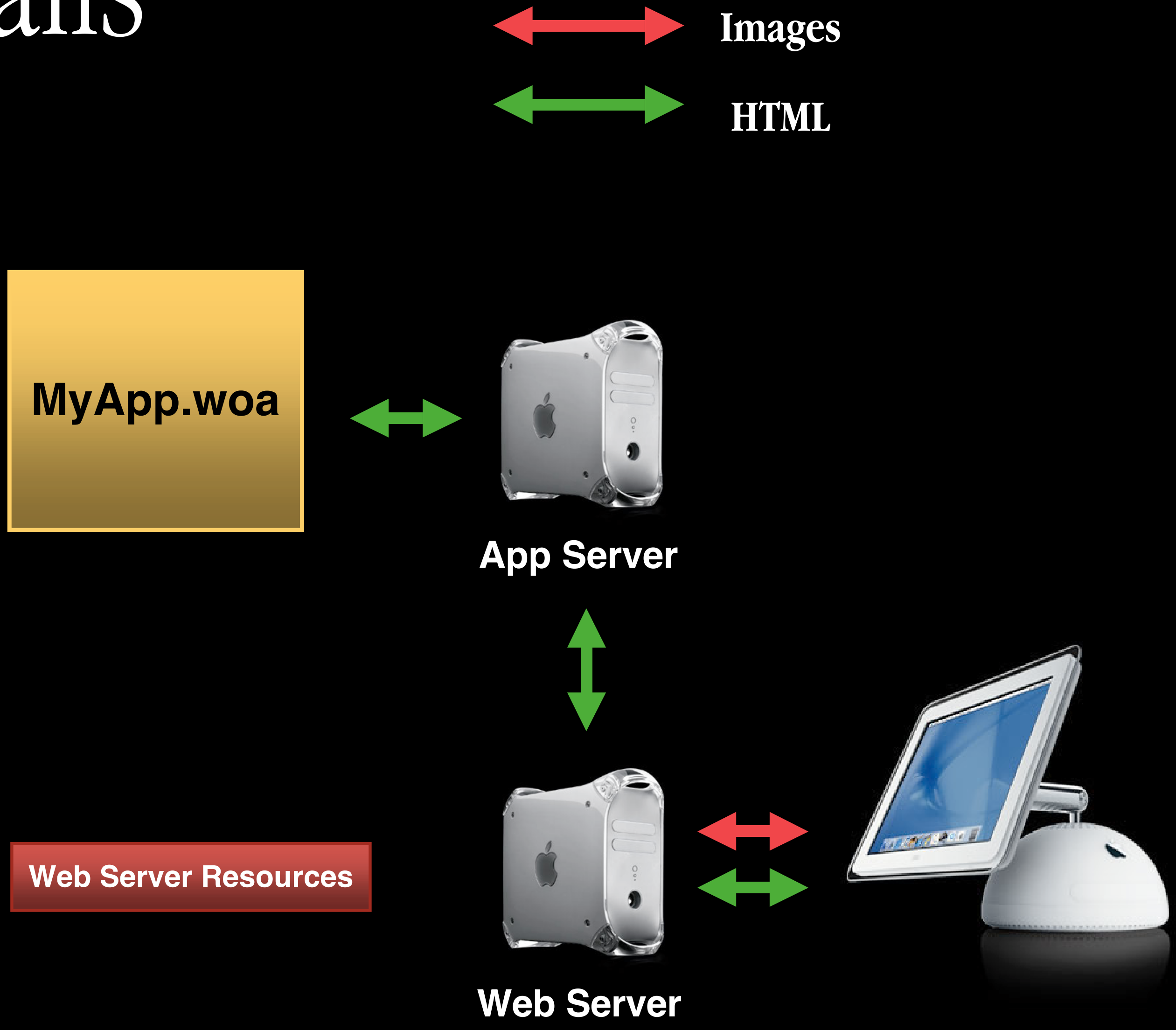
Split Installs



build

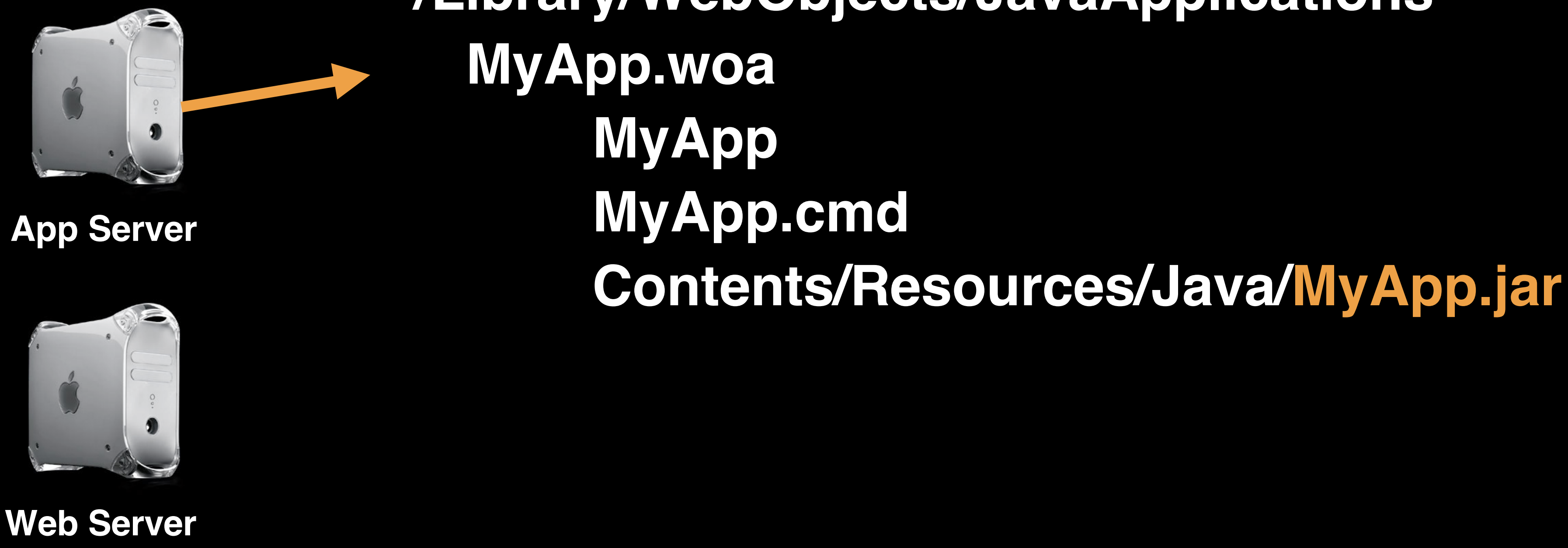
Name

```
▼ MyApp.woa
    ▼ Contents
        Info.plist
        ▼ MacOS
            MacOSClassPath.txt
            MyApp
        pbdevelopment.plist
        PkgInfo
        ▼ Resources
            ▼ Java
                MyApp.jar
            Main.api
            Main.wo
            PageHeader.api
            PageHeader.wo
            ProductCatalog.api
            ProductCatalog.wo
            WOAfile.icns
        ▶ UNIX
        ▼ WebServerResources
            Banner.gif
            Logo.jpg
        ▶ Windows
    MyApp
    MyApp.CMD
```



Split Installs

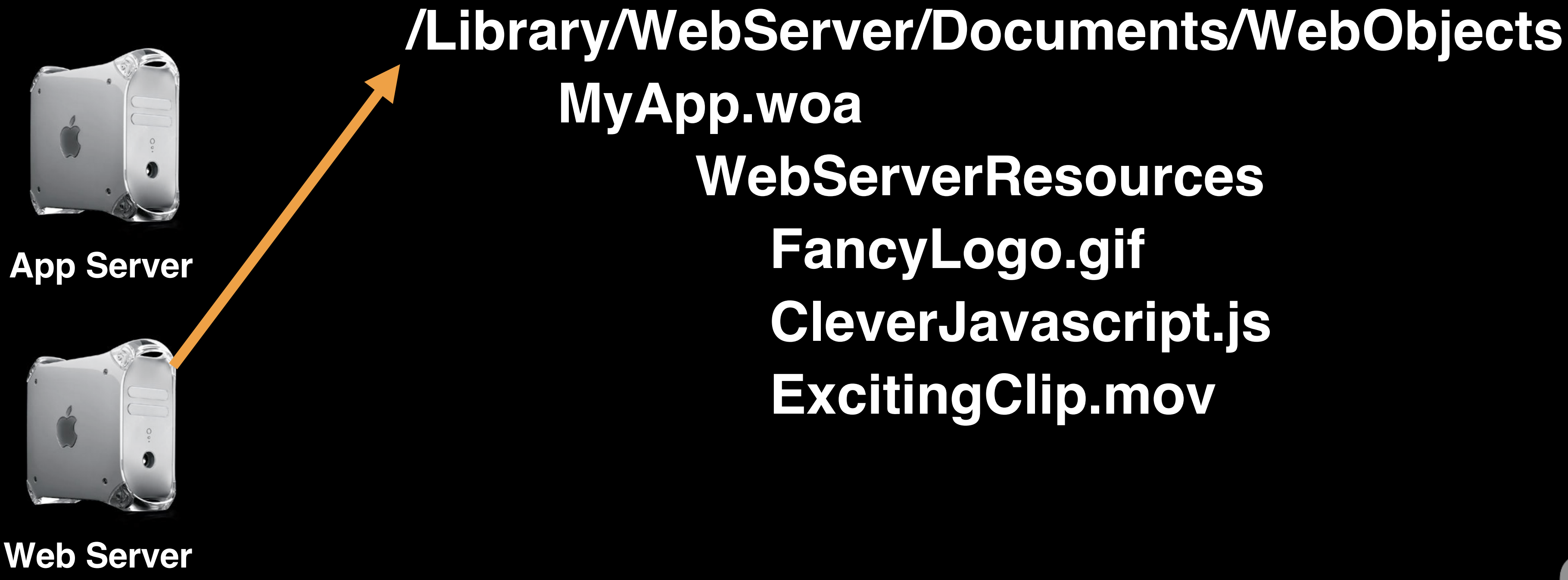
- Actual code goes on the application server, where people can not download it



```
/Library/WebObjects/JavaApplications
    MyApp.woa
         MyApp
         MyApp.cmd
         Contents/Resources/Java/MyApp.jar
```

Split Installs

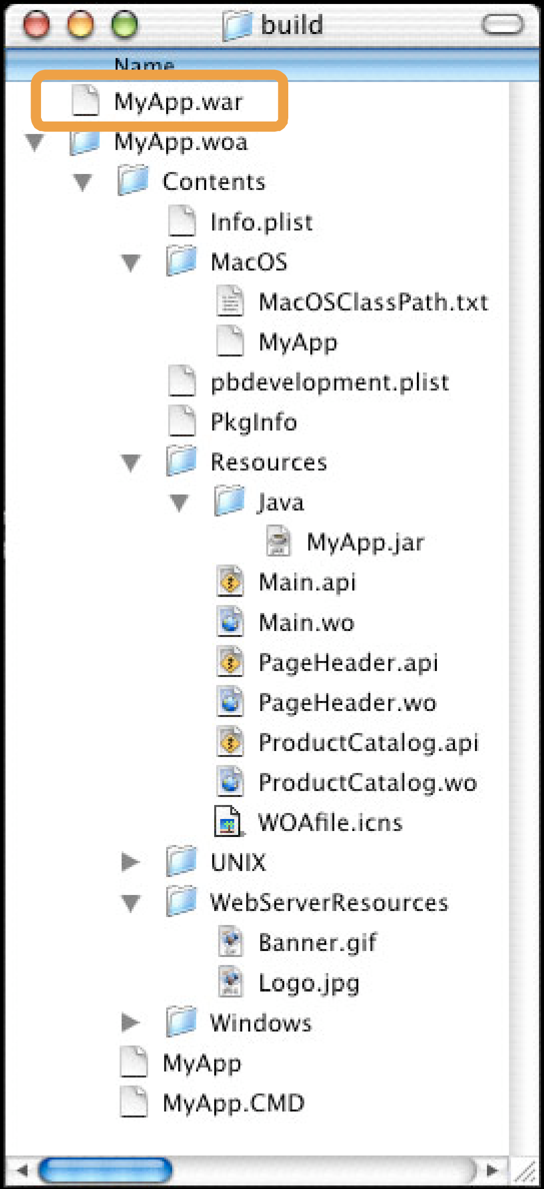
- Images and other nonsensitive resources can go directly on the web server



```
/Library/WebServer/Documents/WebObjects
        MyApp.woa
                WebServerResources
                    FancyLogo.gif
                    CleverJavascript.js
                    ExcitingClip.mov
```

Servlet Deployment

WebObject 5.1



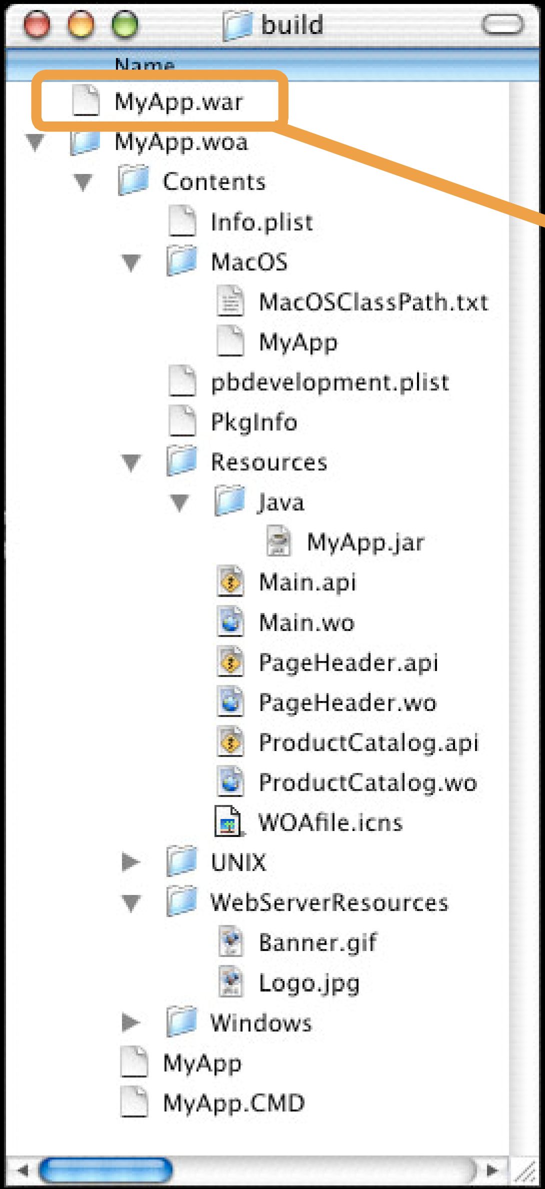
```
MyApp.war
MyApp.woa
  Contents
    Info.plist
    MacOS
      MacOSClassPath.txt
      MyApp
    pbdevelopment.plist
    PkgInfo
    Resources
      Java
        MyApp.jar
      Main.api
      Main.wo
      PageHeader.api
      PageHeader.wo
      ProductCatalog.api
      ProductCatalog.wo
      WOAfile.icns
    UNIX
    WebServerResources
      Banner.gif
      Logo.jpg
    Windows
  MyApp
  MyApp.CMD
```

A bunch of HTML/ WOD components
+ Your own Java code
+ Images and other resources
+ References to frameworks
+JavaWOJSPServlet.framework

All that other stuff,
and a single **MyApp.war** file

Servlet Deployment

WebObject 5.1



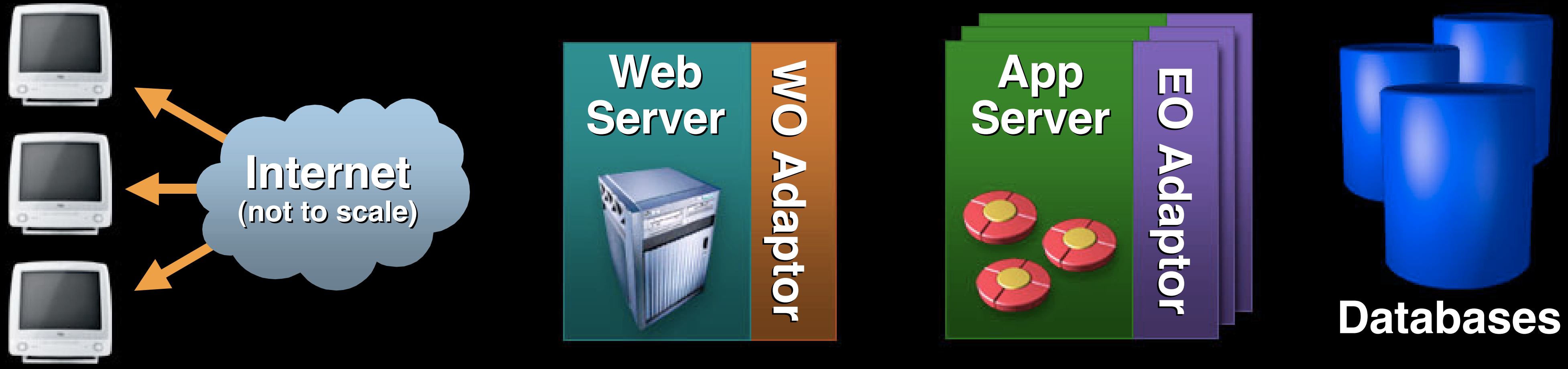
```
build
Name
MyApp.war
▼ MyApp.woa
  ▼ Contents
      Info.plist
    ▼ MacOS
        MacOSClassPath.txt
        MyApp
      pbdevelopment.plist
      PkgInfo
    ▼ Resources
      ▼ Java
          MyApp.jar
        Main.api
        Main.wo
        PageHeader.api
        PageHeader.wo
        ProductCatalog.api
        ProductCatalog.wo
        WOAfile.icns
    ▶ UNIX
    ▼ WebServerResources
        Banner.gif
        Logo.jpg
    ▶ Windows
  MyApp
  MyApp.CMD
```

- Just copy the .war file to the Servlet Container and let it worry about running your app

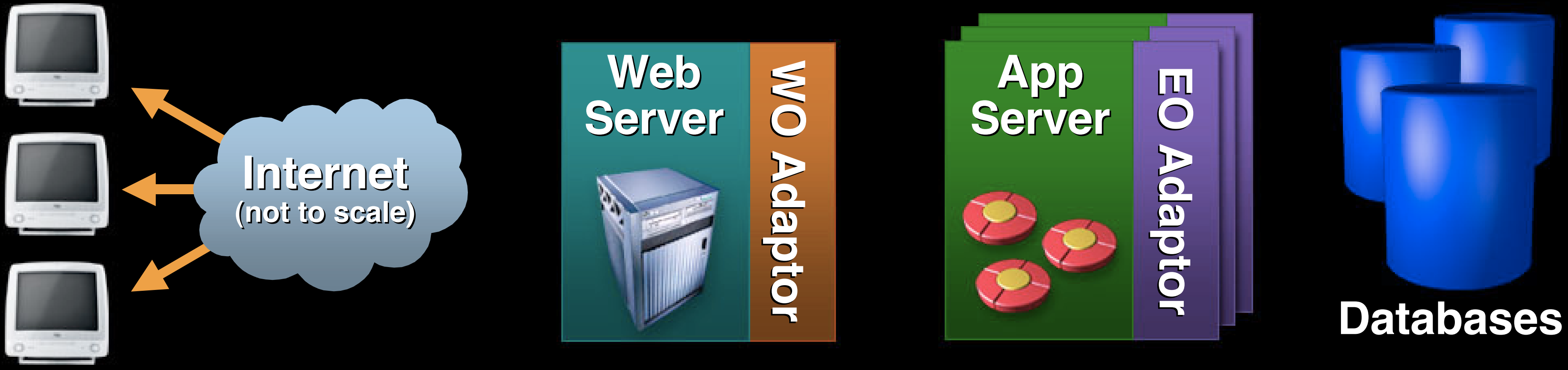
Servlet Container

/Library/Tomcat/webapps/**MyApp.war**

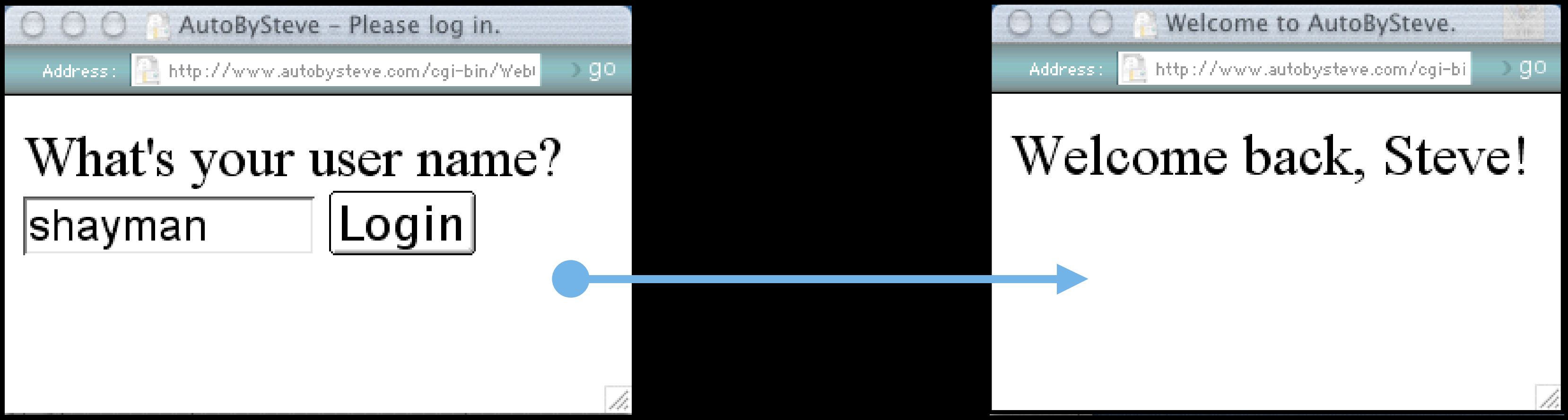
WebObjects in Action



Any OS Any Web Server	Apache Netscape IIS CGI	Mac OS X Server Windows 2000 Solaris Other Java 2 Platforms	JDBC Oracle SQL Server Openbase JNDI LDAP



A really simple app



AutoBySteve – Please log in.

Address: http://www.autobysteve.com/cgi-bin/Webl

What's your user name?

shayman Login

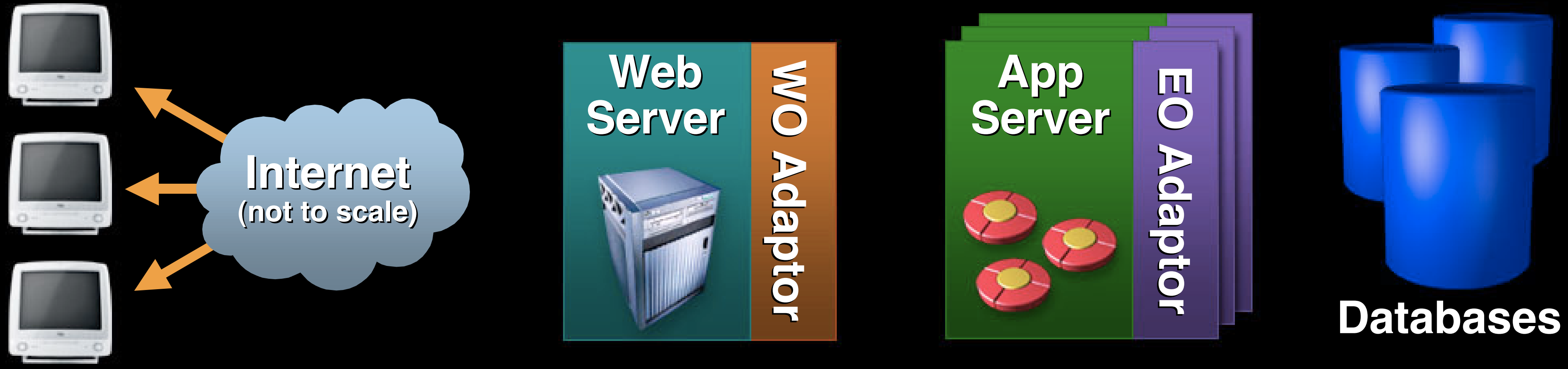
Welcome to AutoBySteve.

Address: http://www.autobysteve.com/cgi-bi

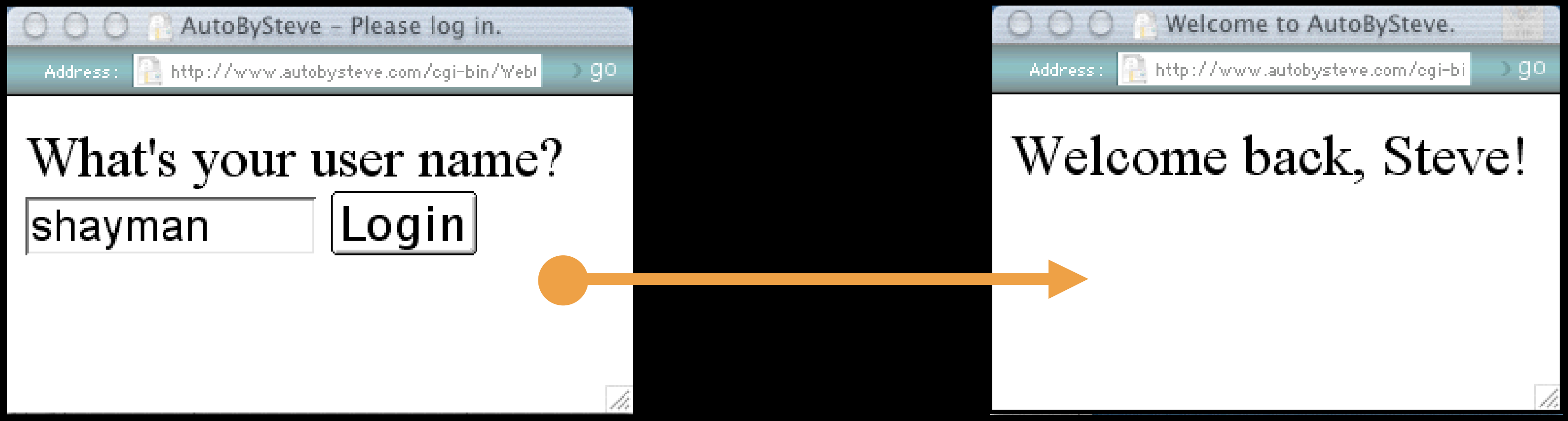
Welcome back, Steve!

1. Process a request

2. Generate a response



3 Steps WO does for you automatically



AutoBySteve - Please log in.

Address: http://www.autobysteve.com/cgi-bin/Webl

What's your user name?

shayman Login

Welcome to AutoBySteve.

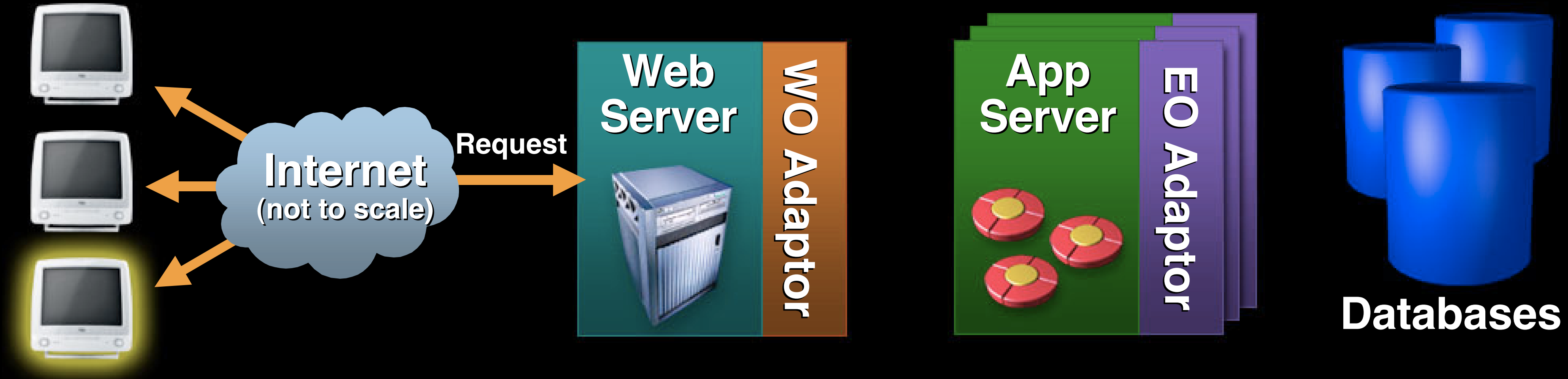
Address: http://www.autobysteve.com/cgi-bi

Welcome back, Steve!

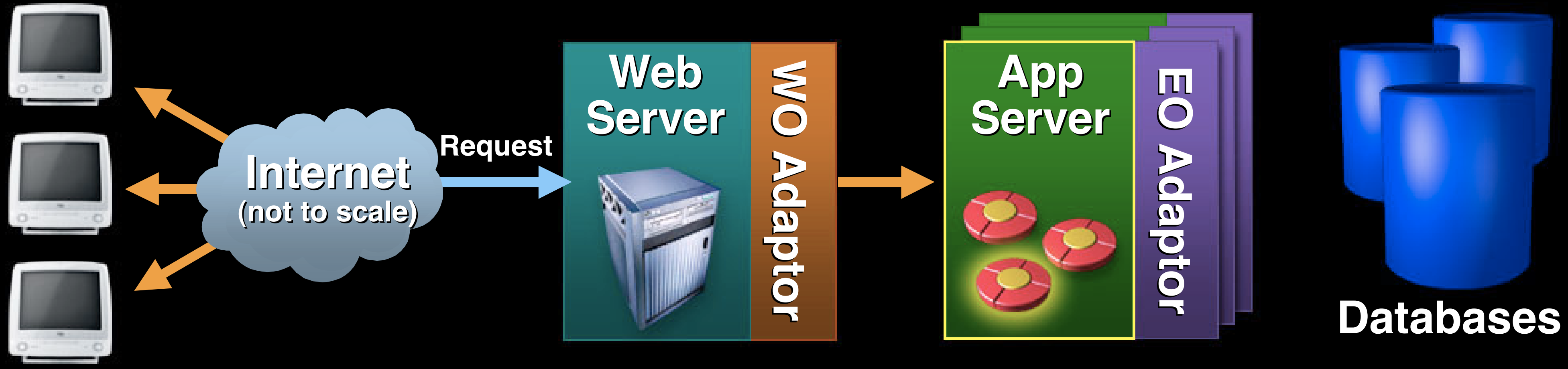
1. Take values from request

3. Generate a response

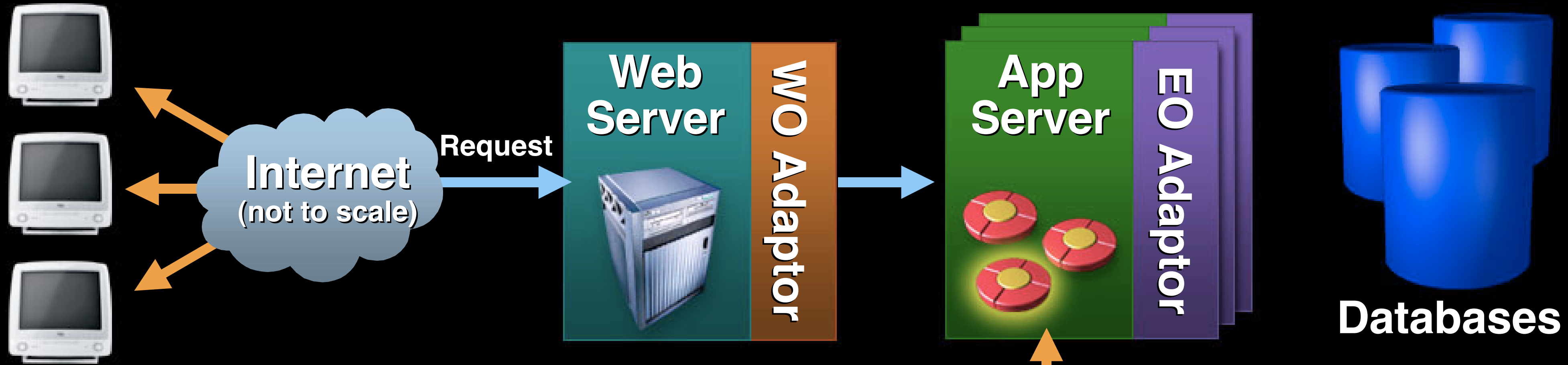
2. Invoke an action



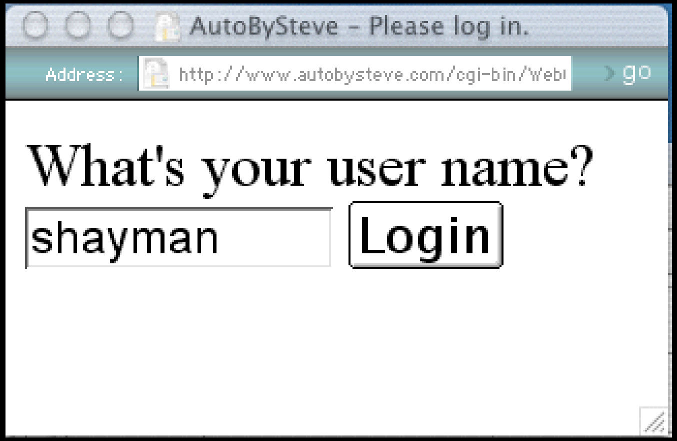
- User submits request to web server



- Web server adaptor finds appropriate application server and instance, and forwards the request to it
  - First request? A random instance is chosen, and a new Session created
  - Subsequently?
    Return to the original one



- Session finds the requesting WOComponent


AutoBySteve - Please log in.
Address:
http://www.autobysteve.com/cgi-bin/Web
go

What's your user name?

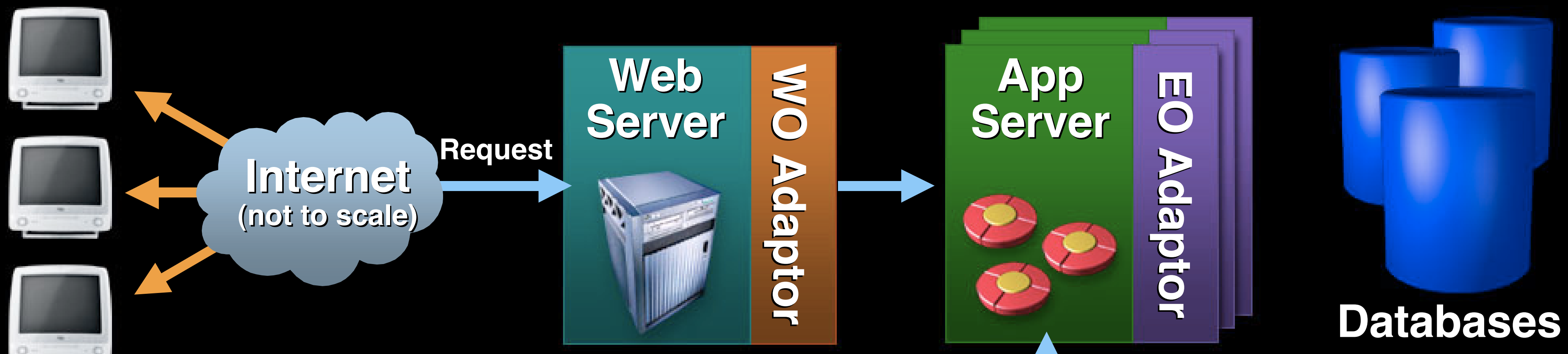
shayman

Login

Login.wo

```
<HTML>
What’s your user name?
<WO NAME=FORM>
<WO NAME=INPUT>
<WO NAME=SUBMIT>
```

```
INPUT: WOTextField
{ value = userName; }
SUBMIT: WOSubmitButton {
  action = handleLogin; }
```

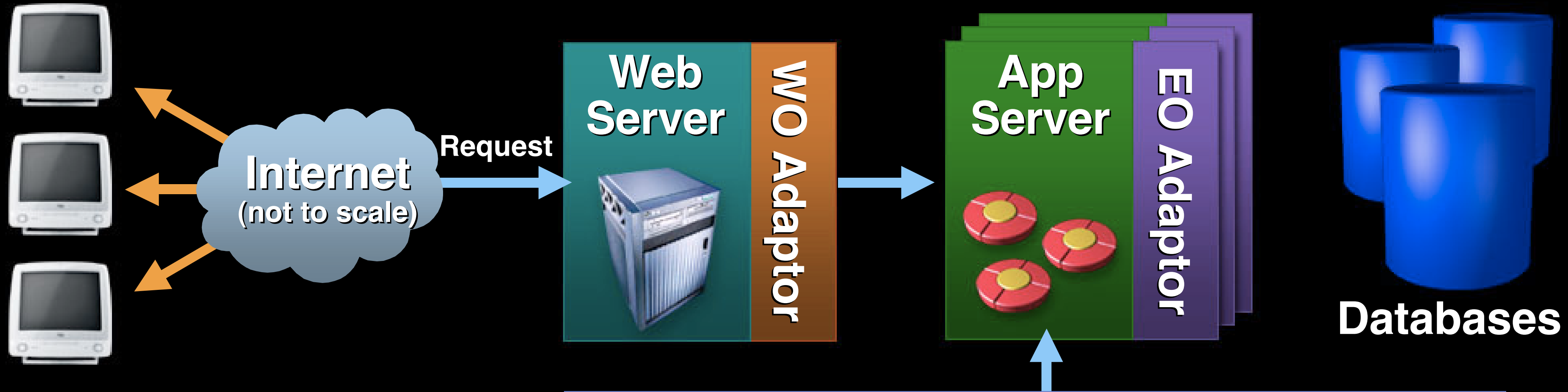


- The requesting component studies any form values that have been submitted

Login.wo

```
<HTML>
What's your user name?
<WO NAME=FORM>
<WO NAME=INPUT>
<WO NAME=SUBMIT>
```

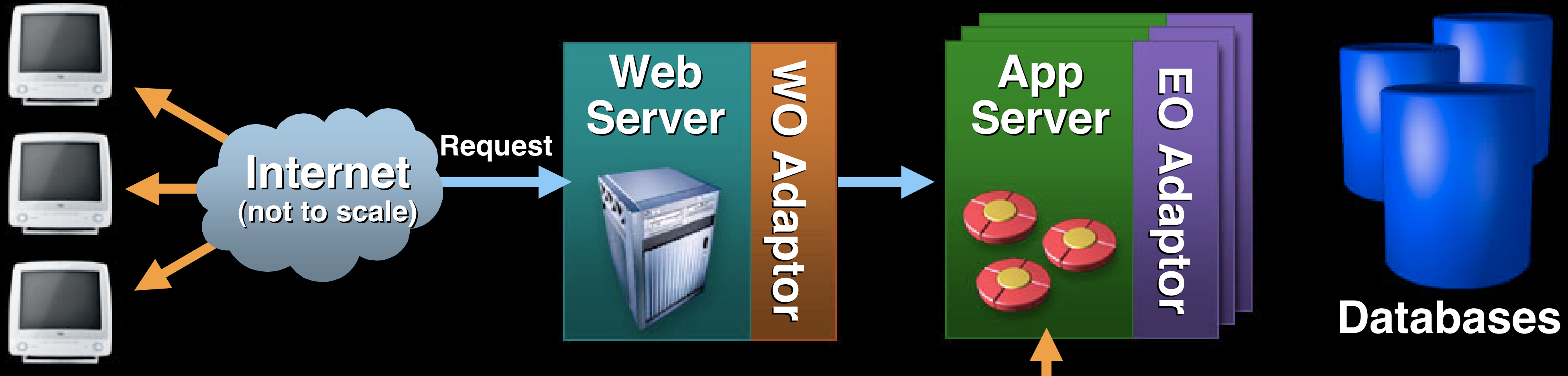
```
INPUT: WOTextField
{ value = userName; }
SUBMIT: WOSubmitButton {
  action = handleLogin; }
```



- Requesting component then figures out what action to invoke
  - Action returns another component

```
//  Login.java
public class Login extends WOComponent {
  public String userName;
  public WOComponent handleLogin() {
            session().fetchCustomer(userName);
            return pageWithName(“Welcome”);
  }
}
```

```
INPUT: WOTextField
{ value = userName; }
SUBMIT: WOSubmitButton {
  action = handleLogin; }
```

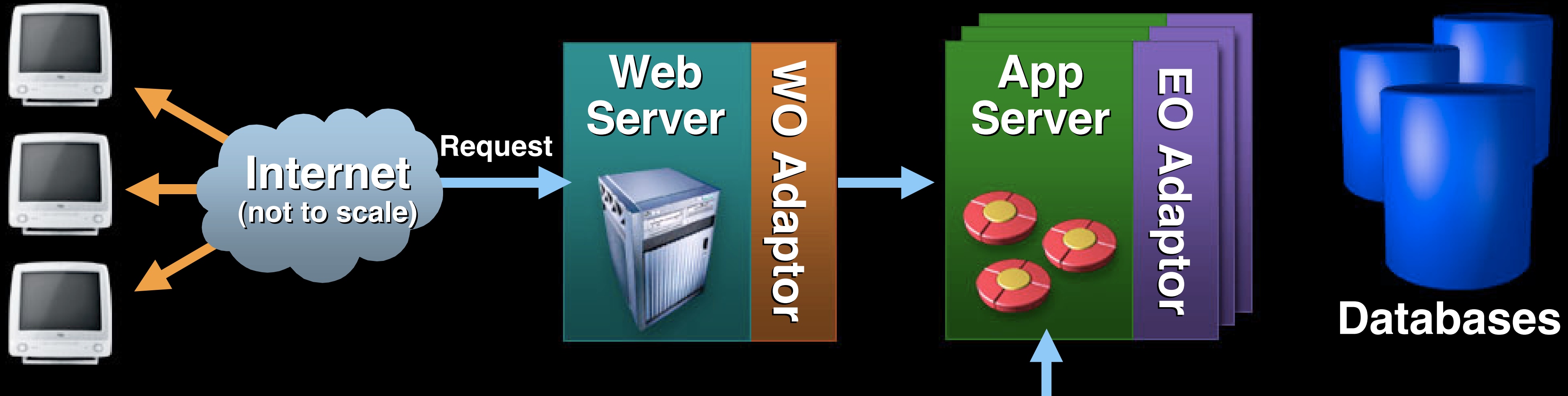


- Action is invoked, and session loads the resulting response component

Welcome.wo

```
<HTML>
Welcome back,
<WEBOBJECT
NAME=UserName>
```

```
UserName: WOString
{ value =
 session.customer.firstName;
}
```

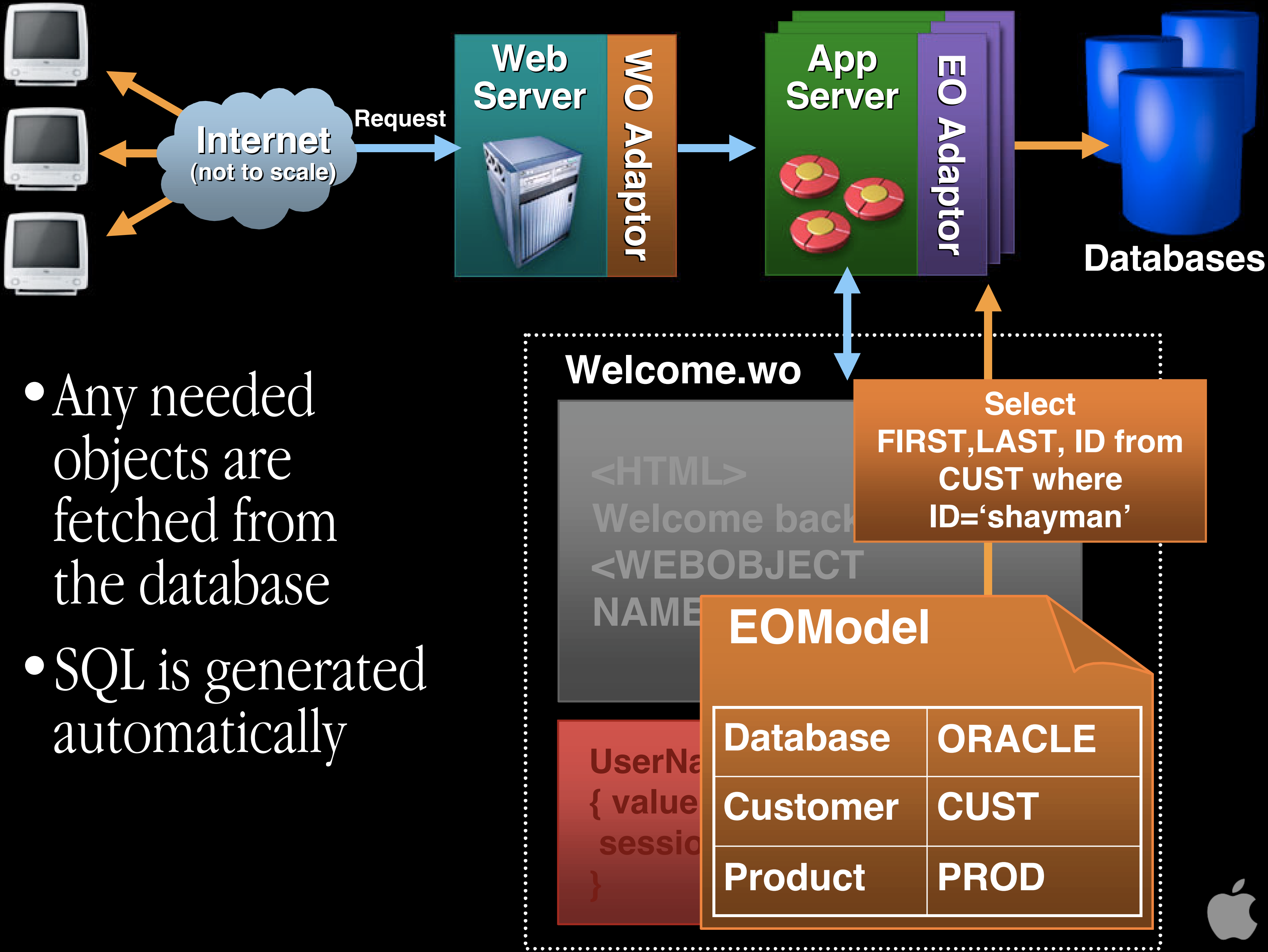


- WEBOBJECT tags and corresponding messages are identified

Welcome.wo

```
<HTML>
Welcome back,
<WEBOBJECT
NAME=UserName>
```

```
UserName: WOString
{ value =
 session.customer.firstName;
}
```

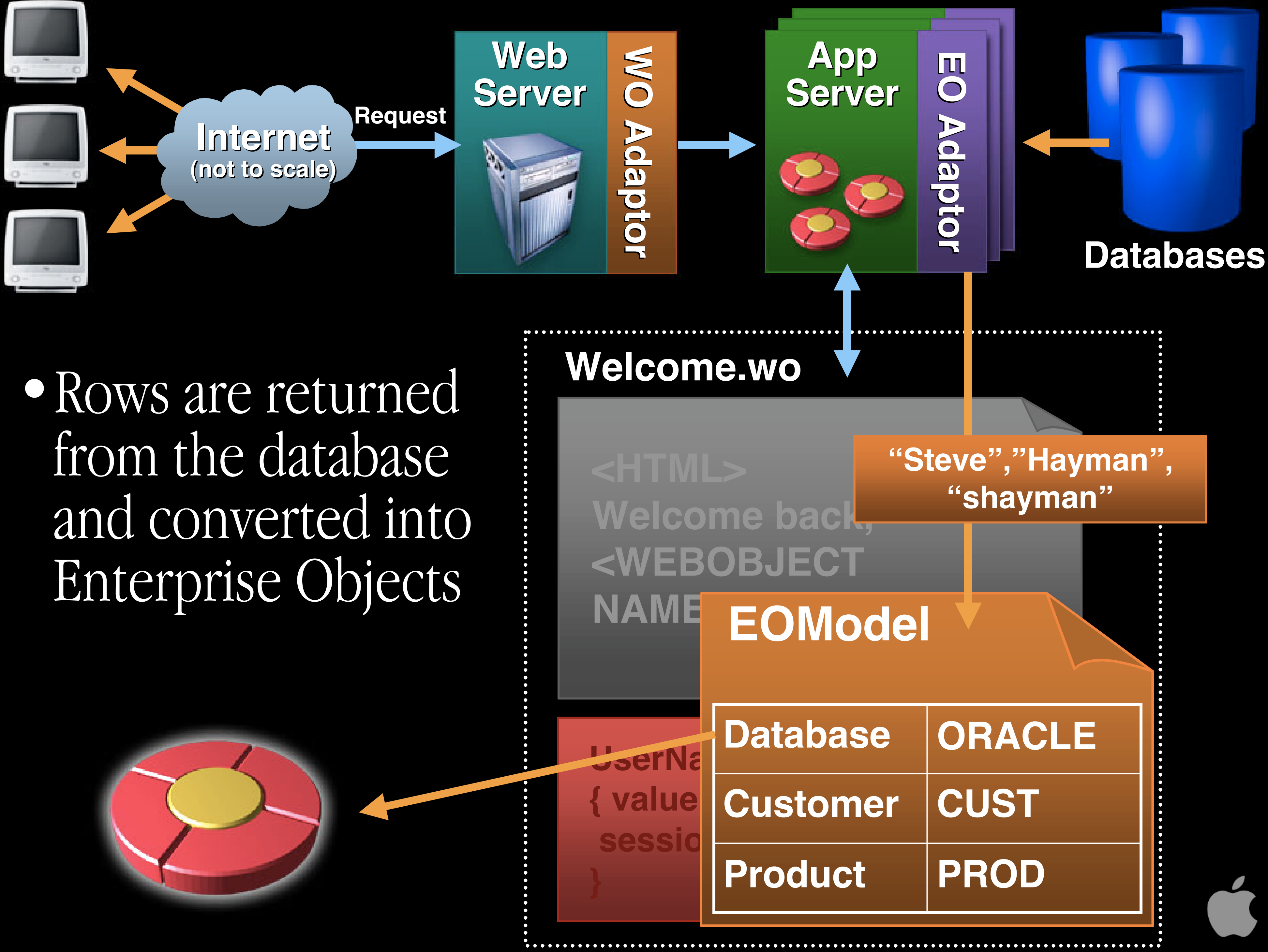


- Any needed objects are fetched from the database
- SQL is generated automatically

Select FIRST,LAST, ID from CUST where ID='shayman'

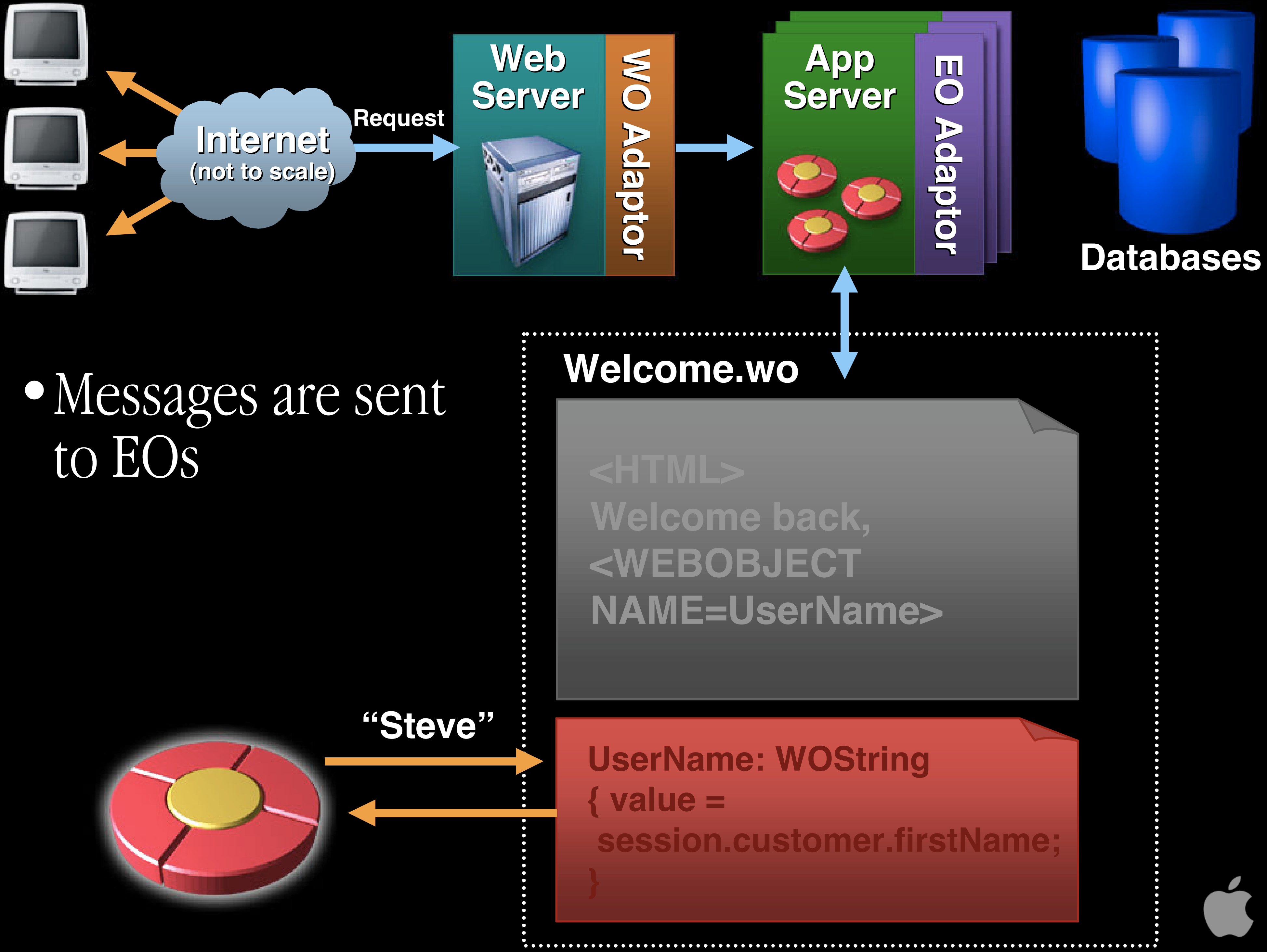
Database	ORACLE
Customer	CUST
Product	PROD

- Rows are returned from the database and converted into Enterprise Objects



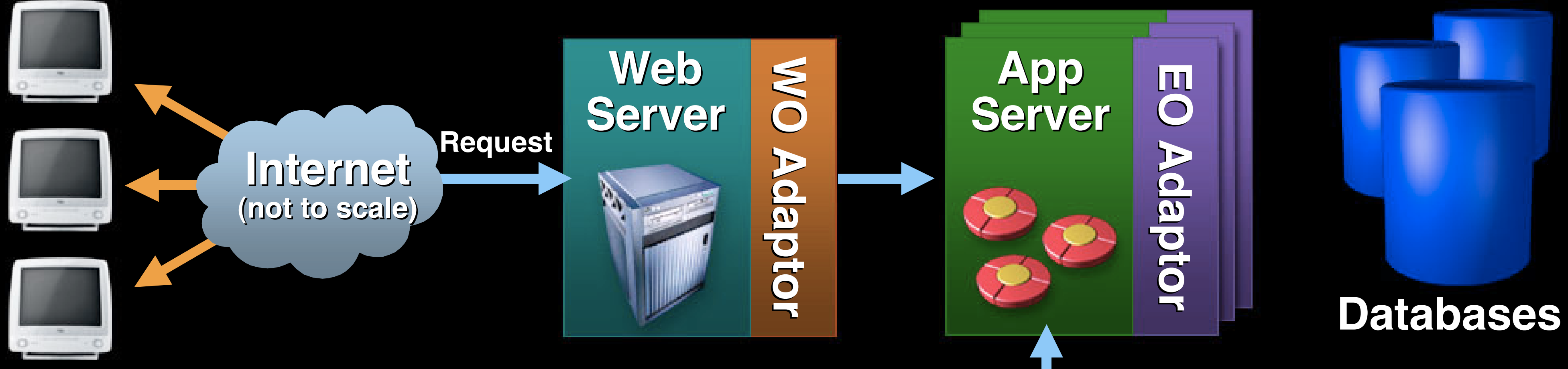
Database	ORACLE
Customer	CUST
Product	PROD

- Messages are sent to EOs

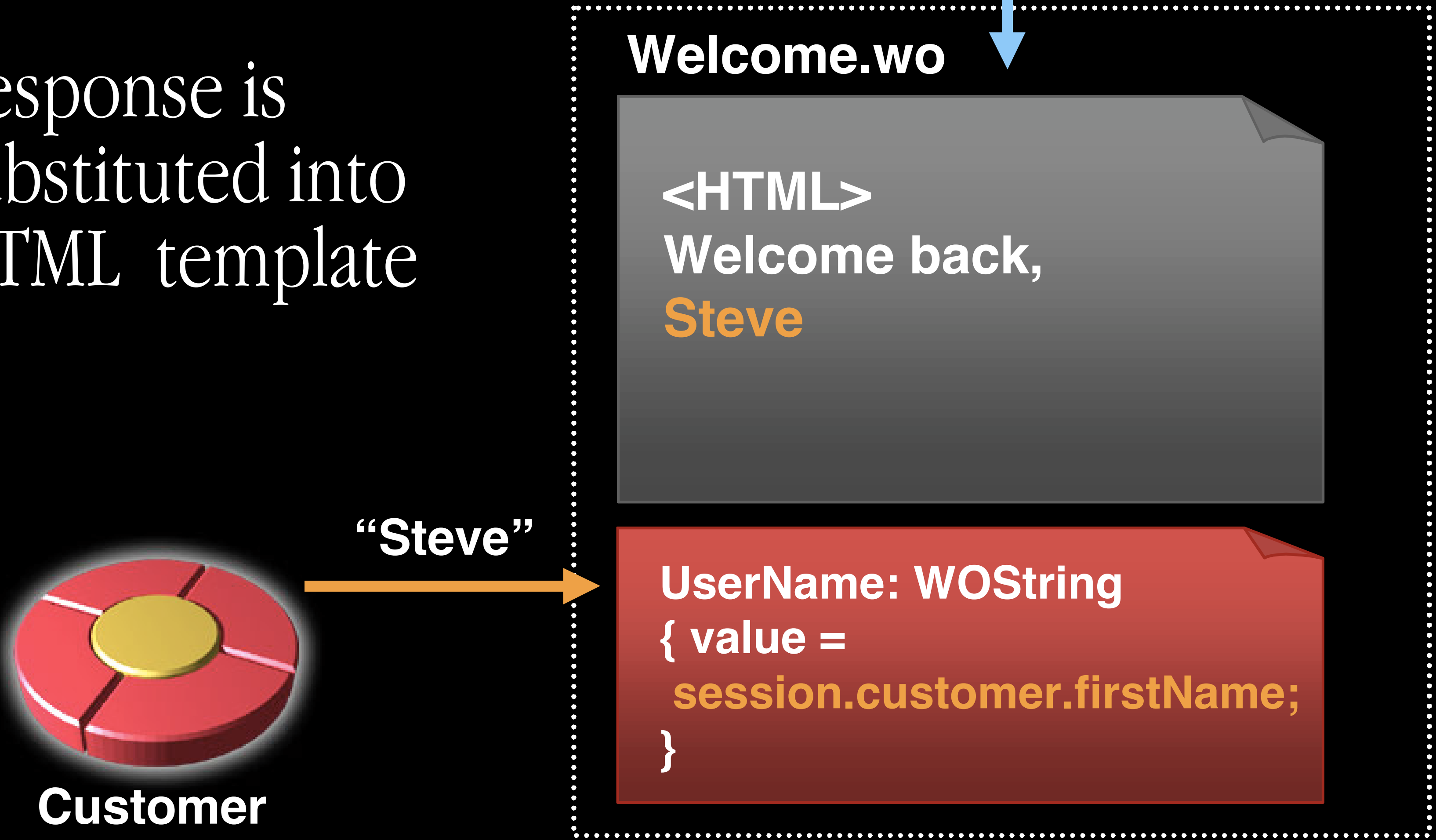


```
<HTML>
Welcome back,
<WEBOBJECT
NAME=UserName>
```

```
UserName: WOString
{ value =
 session.customer.firstName;
}
```

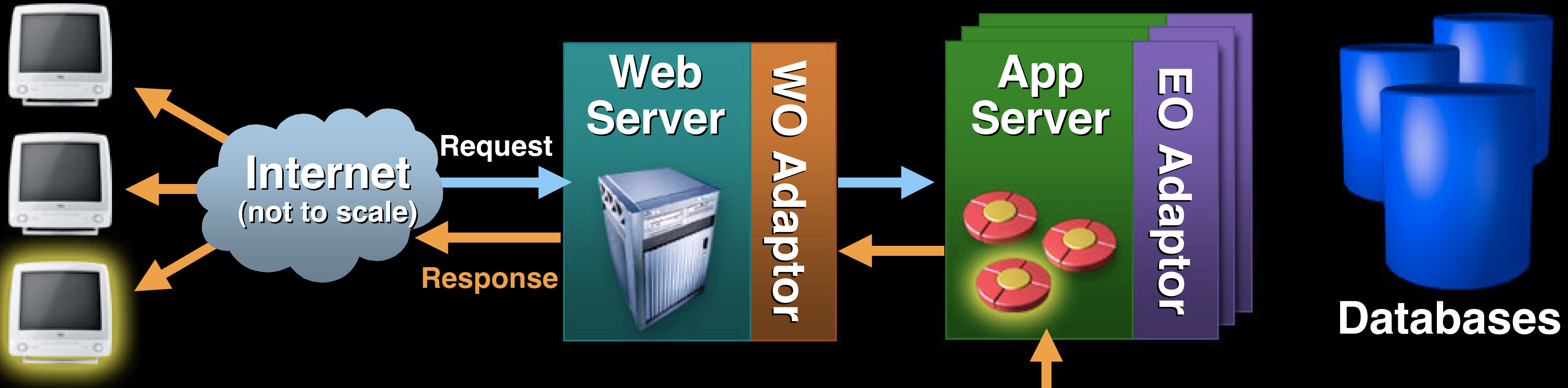


- Response is substituted into HTML template

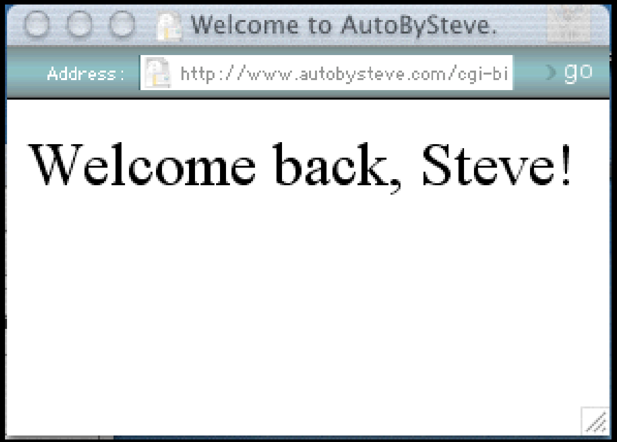


```
<HTML>
Welcome back,
Steve
```

```
UserName: WOString
{ value =
 session.customer.firstName;
}
```



- Session sends plain HTML back to user

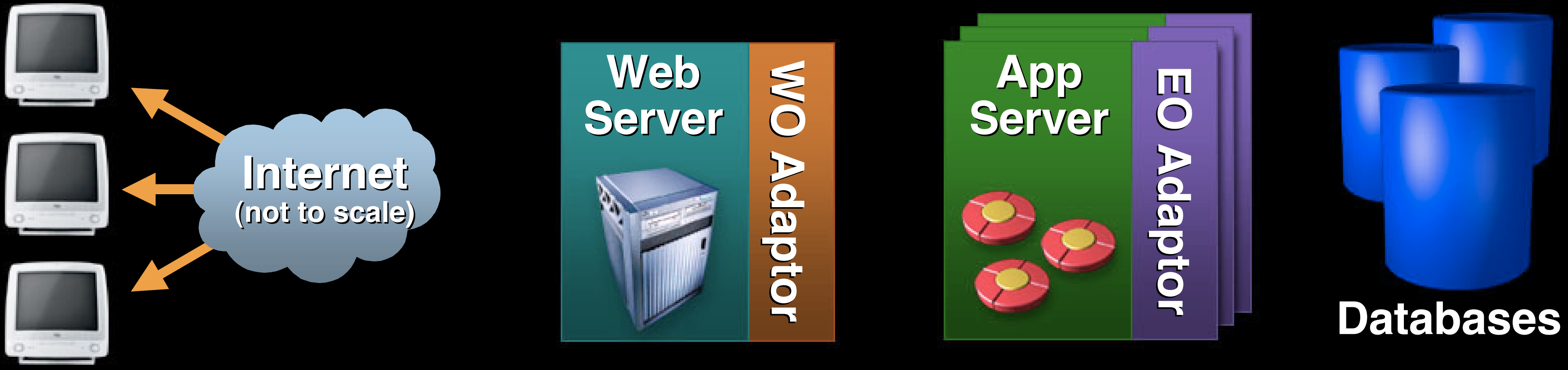


Welcome back, Steve!

Welcome.wo

```
<HTML>
Welcome back,
Steve
```

```
UserName: WOString
{ value =
 session.customer.firstName;
}
```



- That’s the Request/Response Loop

```
public void takeValuesFromRequest   ( WORequest aRequest,
                                      WOContext aContext )
public WOElement invokeAction       ( WORequest aRequest,
                                      WOContext aContext )
public void appendToResponse        ( WOResponse aResponse,
                                      WOContext aContext )
```

- Application, Session, and all Components participate (If they want to)

Please Note!

- No mixing of SQL and HTML
- Database, interface, business logic are all completely separate and independent
- No need to write SQL or HTML at all (Unless you want to)

The Tools

WebObjects Builder

You’ll Create WO Components

Main.wo/

- Main.html
- Main.wod

```
<HTML>
This product
<WEBOBJECT NAME=Image4>
costs
<WEBOBJECT NAME=String1>
<HR>
<WEBOBJECT NAME=Link1>
   Click here
</WEBOBJECT>   to order.
```

```
Image4: WOImage {
   src = “pic.gif”; }
String1: WOString {
  value = product.cost
  numberFormat =
        “$#,###”;}
Link1: WOHyperlink {
   action = placeOrder;  }
```

Main.java

```
public class Order extends WOComponent {
  private ShoppingItem product;
  public WOComponent placeOrder( ) {
    session().addToShoppingCart(product);
    return pageWithName(“Checkout”);
  }
}
```

Or, in 5.1—JSP Style

Main.jsp

```
<HTML>
<%@ taglib uri="/WOtaglib_1_0.tld" prefix="wo" %>
<% WOServletAdaptor.initStatics(application); %>
<%@ page import = "com.webobjects.jspservlet.*" %>
<% NSMutableArray cart = new NSMutableArray(); %>
This product
<wo:component className="MyImageComponent">
   <wo:binding key="filename" value='<%= ”pic.gif" %>' />
</wo:component>
costs
<wo:component className=“WOString”>
             <wo:binding key=“value” value=“product.cost”/>
             <wo:binding key=“numberFormat” value=“$#,###” />
</wo:component>
<HR>
<wo:component className=“WOHyperlink”>
             <wo:binding key=“action” value=“placeOrder”/>
Click here
</wo:component>
 to order.
```

You’ll Create Lots of WO Components

- One for each “page”
- Subcomponents for headers, footers, navigation bars
- Reusable components for common UI elements
- JavaScript client-side tricks
- Components that contain other components

So How Do You Make Those Components?

- They are just text files
- Use whatever you like
    - emacs
    - vi
    - stickies
    - cat

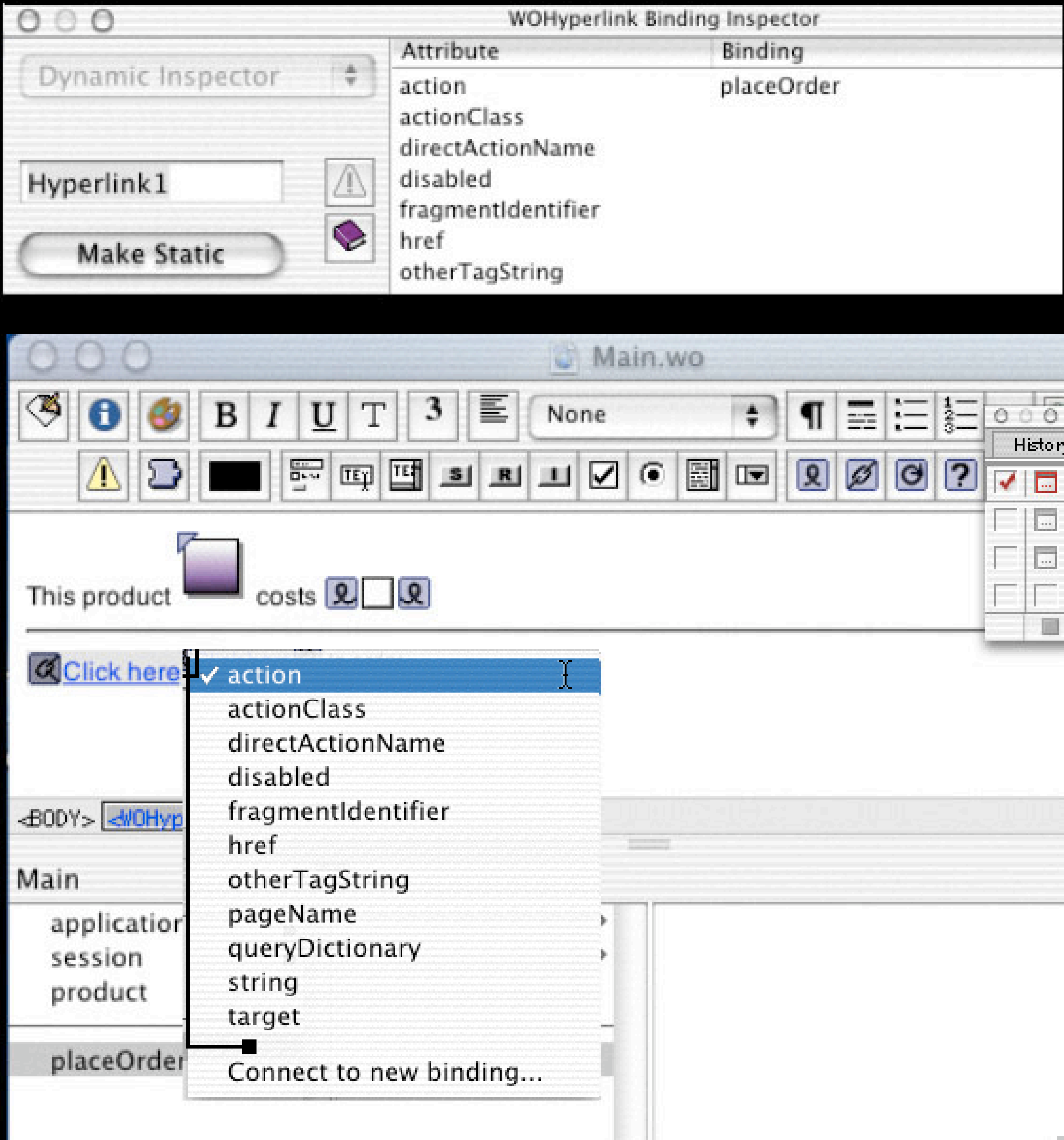
Oh, Come On

- OK. You can use any WYSIWIG HTML editor
- DreamWeaver
- GoLive
- Or even WebObjects Builder

Inspector

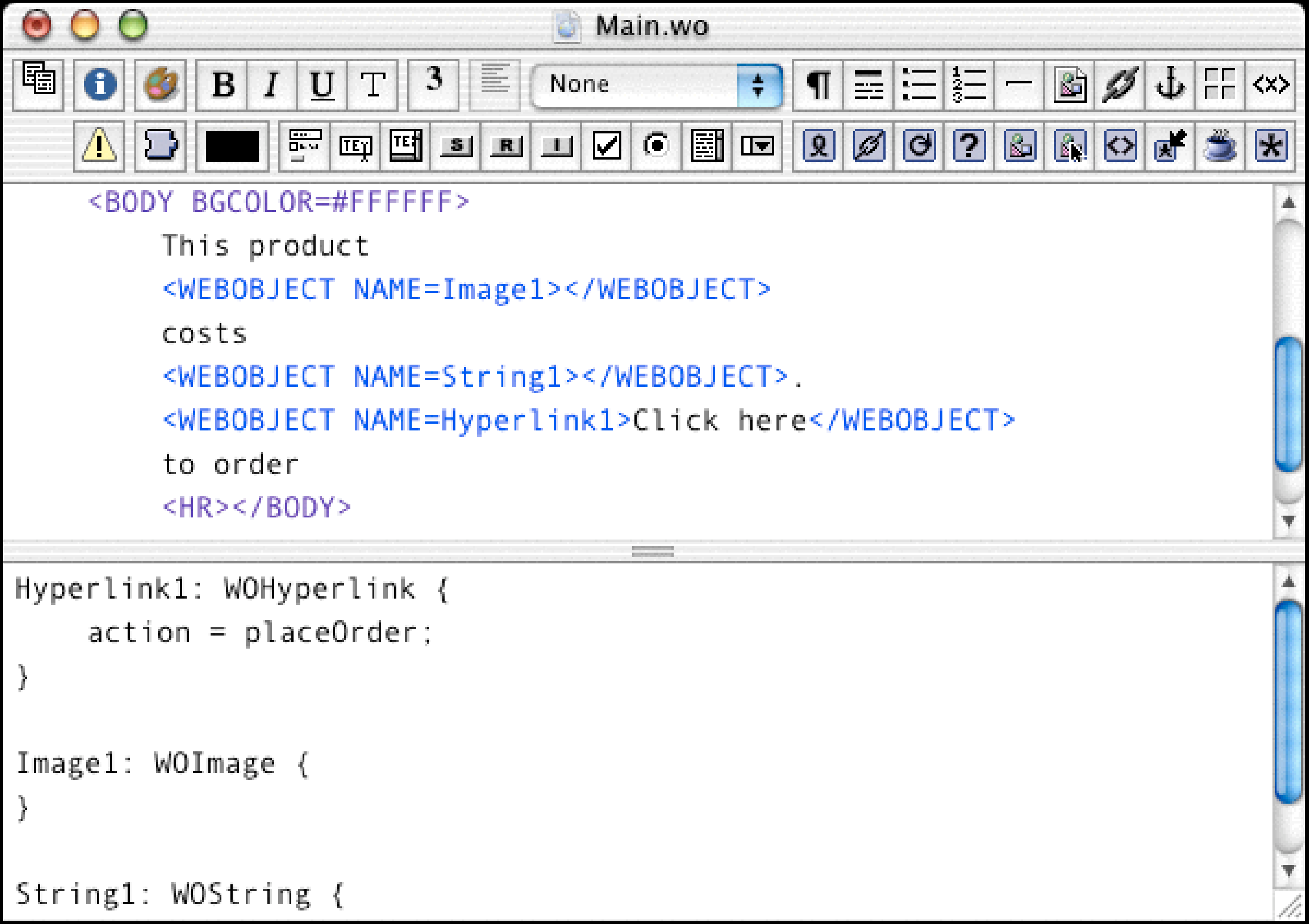
HTML

Objects
and Actions



Attribute	Binding
action	placeOrder
actionClass	
directActionName	
disabled	
fragmentIdentifier	
href	
otherTagString	

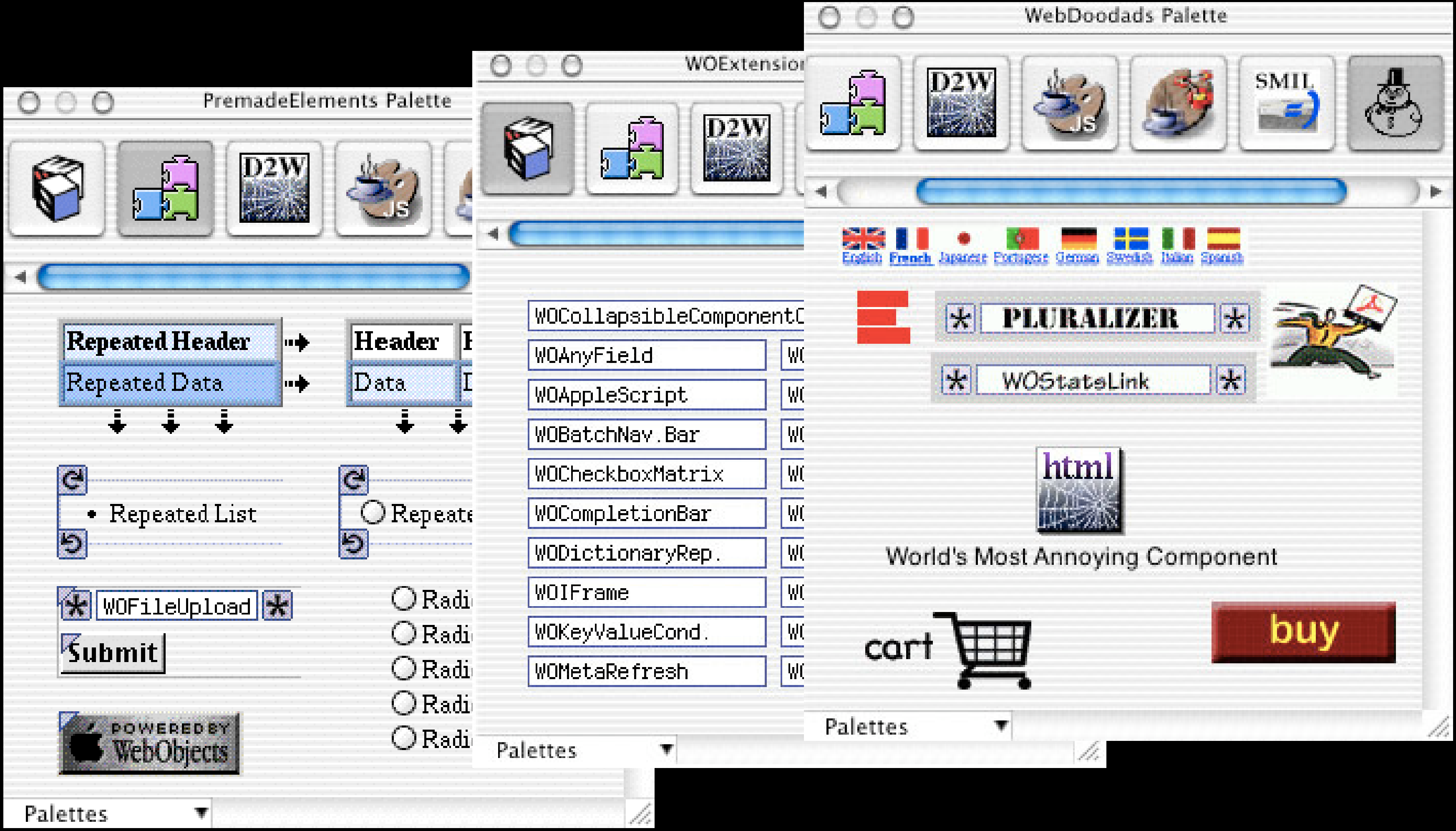
Use WOB in Raw Mode, If You Like



```
    <BODY BGCOLOR=#FFFFFF>
        This product
        <WEBOBJECT NAME=Image1></WEBOBJECT>
        costs
        <WEBOBJECT NAME=String1></WEBOBJECT>.
        <WEBOBJECT NAME=Hyperlink1>Click here</WEBOBJECT>
        to order
        <HR></BODY>
```

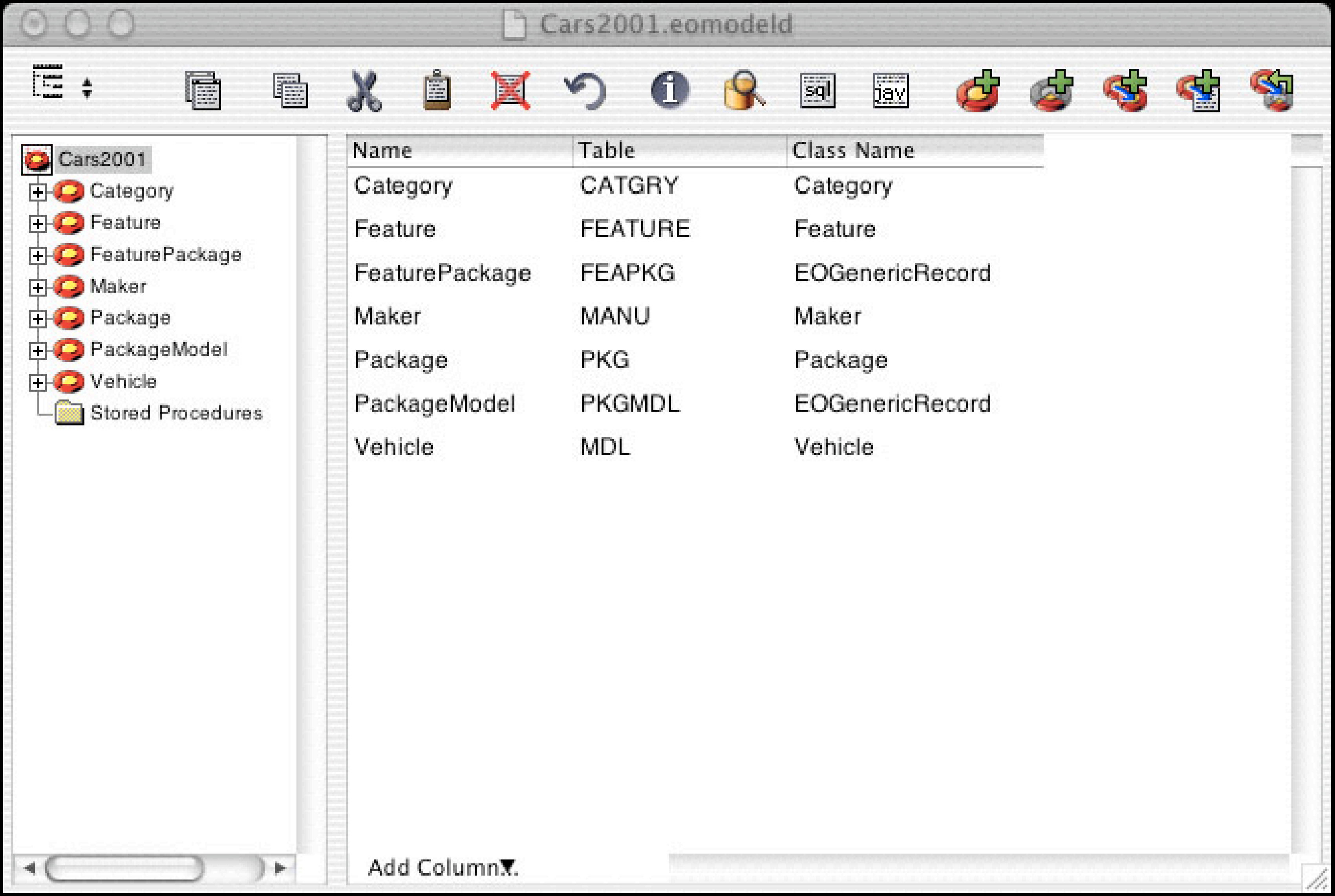
```
Hyperlink1: WOHyperlink {
    action = placeOrder;
}

Image1: WOImage {
}

String1: WOString {
```

Use Palettes to Organize Your Stuff



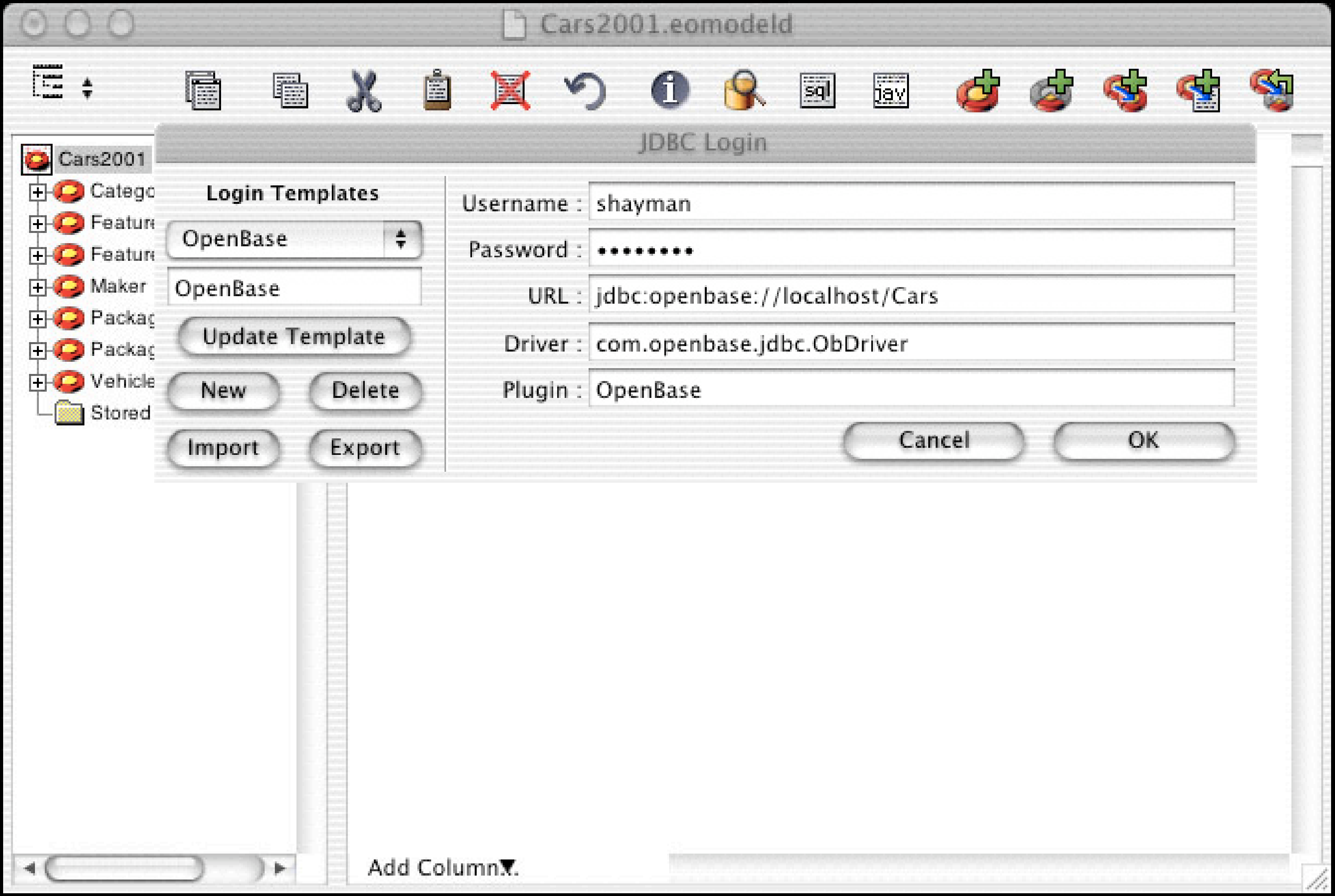
EOModeler

An Editor For .eomodeld Files



Name	Table	Class Name
Category	CATGRY	Category
Feature	FEATURE	Feature
FeaturePackage	FEAPKG	EOGenericRecord
Maker	MANU	Maker
Package	PKG	Package
PackageModel	PKGMDL	EOGenericRecord
Vehicle	MDL	Vehicle

Database Connection Info . . .



Login Templates: OpenBase

OpenBase

Username : shayman

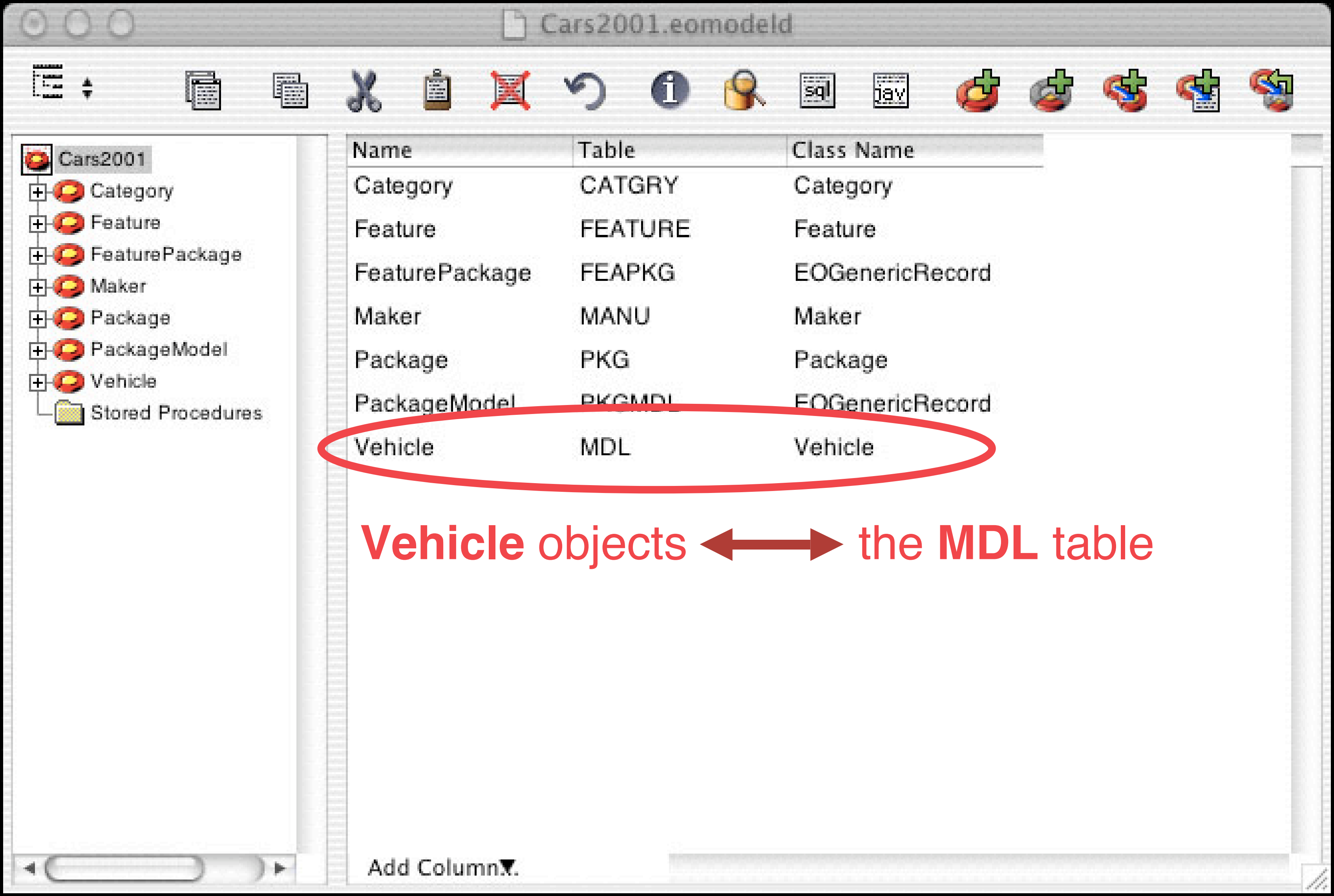
Password : •••••••••

URL : jdbc:openbase://localhost/Cars

Driver : com.openbase.jdbc.ObDriver

Plugin : OpenBase

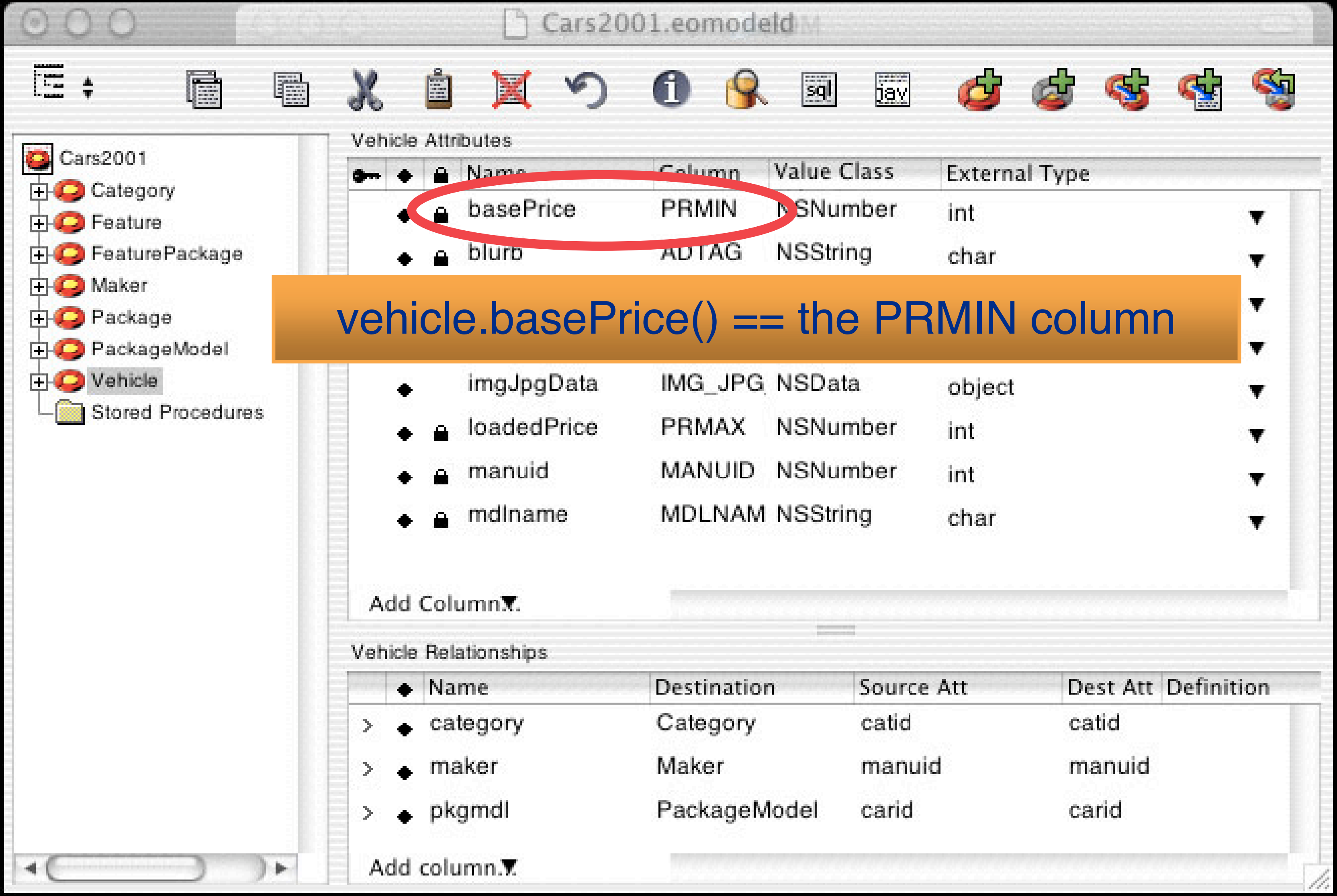
Entities and Tables



Name	Table	Class Name
Category	CATGRY	Category
Feature	FEATURE	Feature
FeaturePackage	FEAPKG	EOGenericRecord
Maker	MANU	Maker
Package	PKG	Package
PackageModel	PKGMDL	EOGenericRecord
Vehicle	MDL	Vehicle

Vehicle objects ⟷ the **MDL** table

Entity Information



Vehicle Attributes

	◆	🔒	Name	Column	Value Class	External Type
	◆	🔒	basePrice	PRMIN	NSNumber	int
	◆	🔒	blurb	ADTAG	NSString	char
	◆		imgJpgData	IMG_JPG	NSData	object
	◆	🔒	loadedPrice	PRMAX	NSNumber	int
	◆	🔒	manuid	MANUID	NSNumber	int
	◆	🔒	mdlname	MDLNAM	NSString	char

Add Column

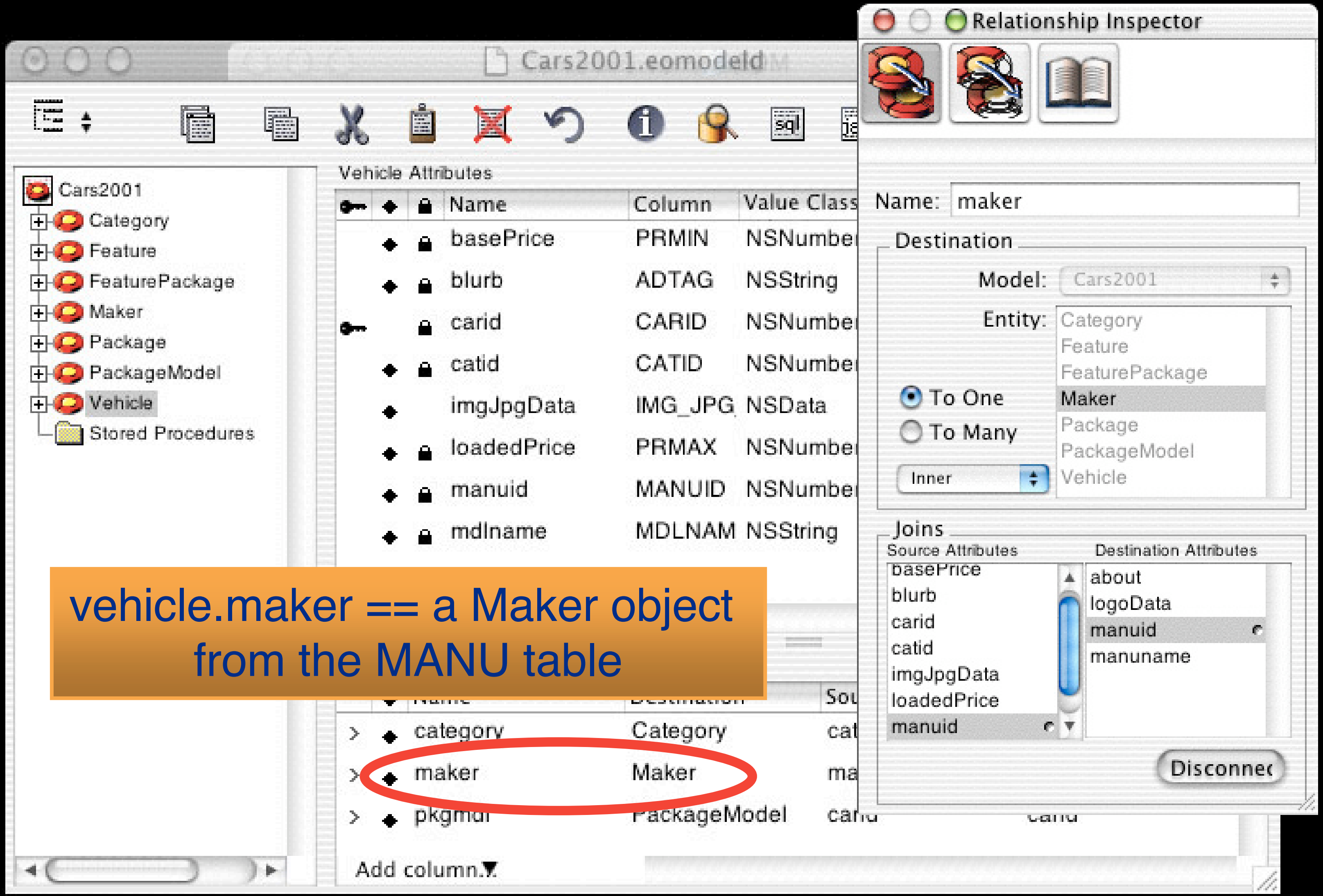
vehicle.basePrice() == the PRMIN column

Vehicle Relationships

	◆	Name	Destination	Source Att	Dest Att	Definition
>	◆	category	Category	catid	catid	
>	◆	maker	Maker	manuid	manuid	
>	◆	pkgmdl	PackageModel	carid	carid	

Add column

Relationship Information



Cars2001.eomodeld

Cars2001
- Category
- Feature
- FeaturePackage
- Maker
- Package
- PackageModel
- Vehicle
- Stored Procedures

Vehicle Attributes

Name	Column	Value Class
basePrice	PRMIN	NSNumbe
blurb	ADTAG	NSString
carid	CARID	NSNumbe
catid	CATID	NSNumbe
imgJpgData	IMG_JPG	NSData
loadedPrice	PRMAX	NSNumbe
manuid	MANUID	NSNumbe
mdlname	MDLNAM	NSString

Name	Destination	Sou
category	Category	cat
maker	Maker	ma
pkgmdl	PackageModel	carid

Add column

Relationship Inspector

Name: maker

Destination

Model: Cars2001

Entity: Category, Feature, FeaturePackage, Maker, Package, PackageModel, Vehicle

● To One
○ To Many

Inner

Joins

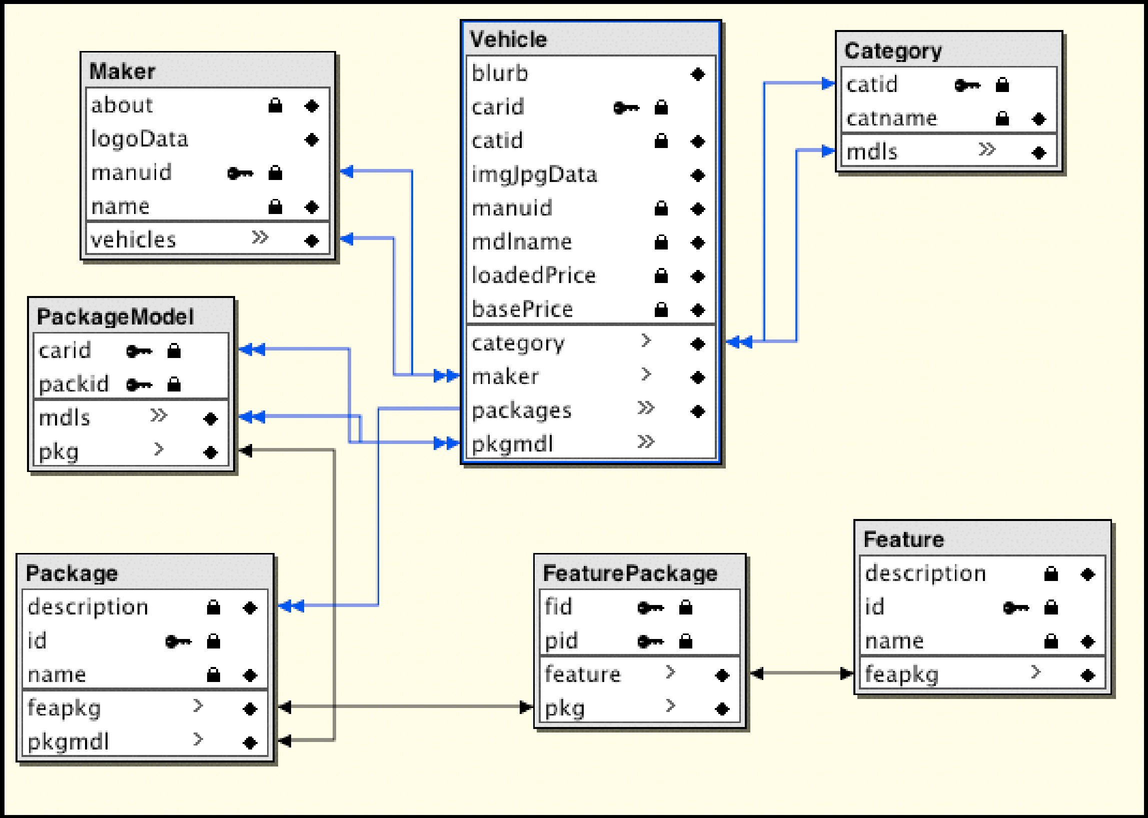
Source Attributes	Destination Attributes
basePrice	about
blurb	logoData
carid	manuid
catid	manuname
imgJpgData	
loadedPrice	
manuid	

Disconnec

vehicle.maker == a Maker object from the MANU table

EOModeler

- Model complex relationships between objects too

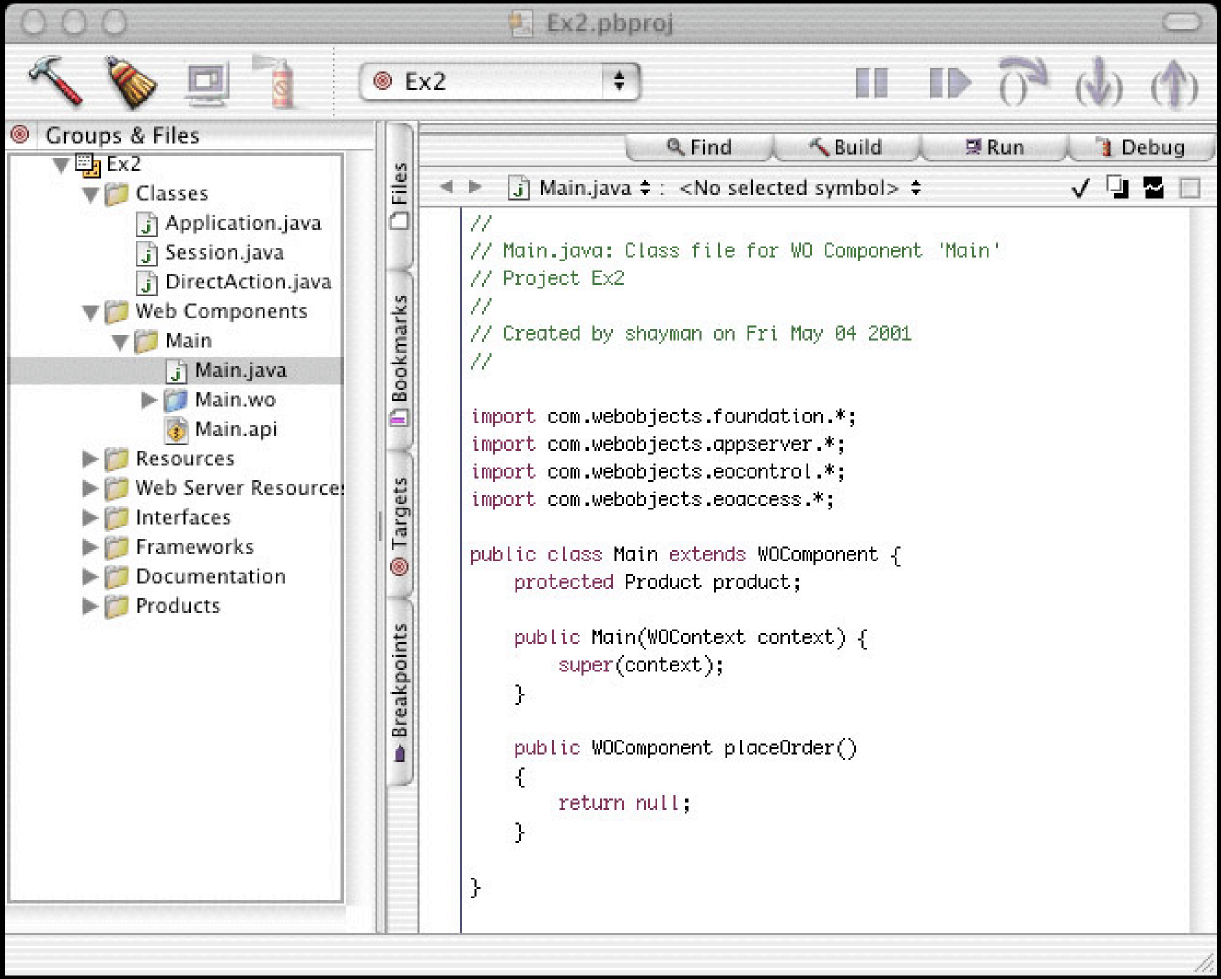


vehicle.maker.name
vehicle.packages
maker.vehicles.@avg.basePrice

Project Builder

Project Builder

- Coordinates resources
- Code editing and debugging
- Localization



```
//
// Main.java: Class file for WO Component 'Main'
// Project Ex2
//
// Created by shayman on Fri May 04 2001
//

import com.webobjects.foundation.*;
import com.webobjects.appserver.*;
import com.webobjects.eocontrol.*;
import com.webobjects.eoaccess.*;

public class Main extends WOComponent {
    protected Product product;

    public Main(WOContext context) {
        super(context);
    }

    public WOComponent placeOrder()
    {
        return null;
    }

}
```

Demo

Now Let’s Build Something

Superduper Assistants

Got a model? You are 95% done.

Direct to Web

Your EOModel

+ Direct to Web

A complete, full featured, and good-looking web app

Direct to Java Client

Your EOModel

+ Direct to Java Client

A complete, full featured, and good-looking Java-based application

Demo

Direct to Web
Direct to Java Client

Demo of Servlet Deployment

- Just drag/drop the .war to the right place and restart tomcat
- Provided I remembered to check the right checkbox at the beginning

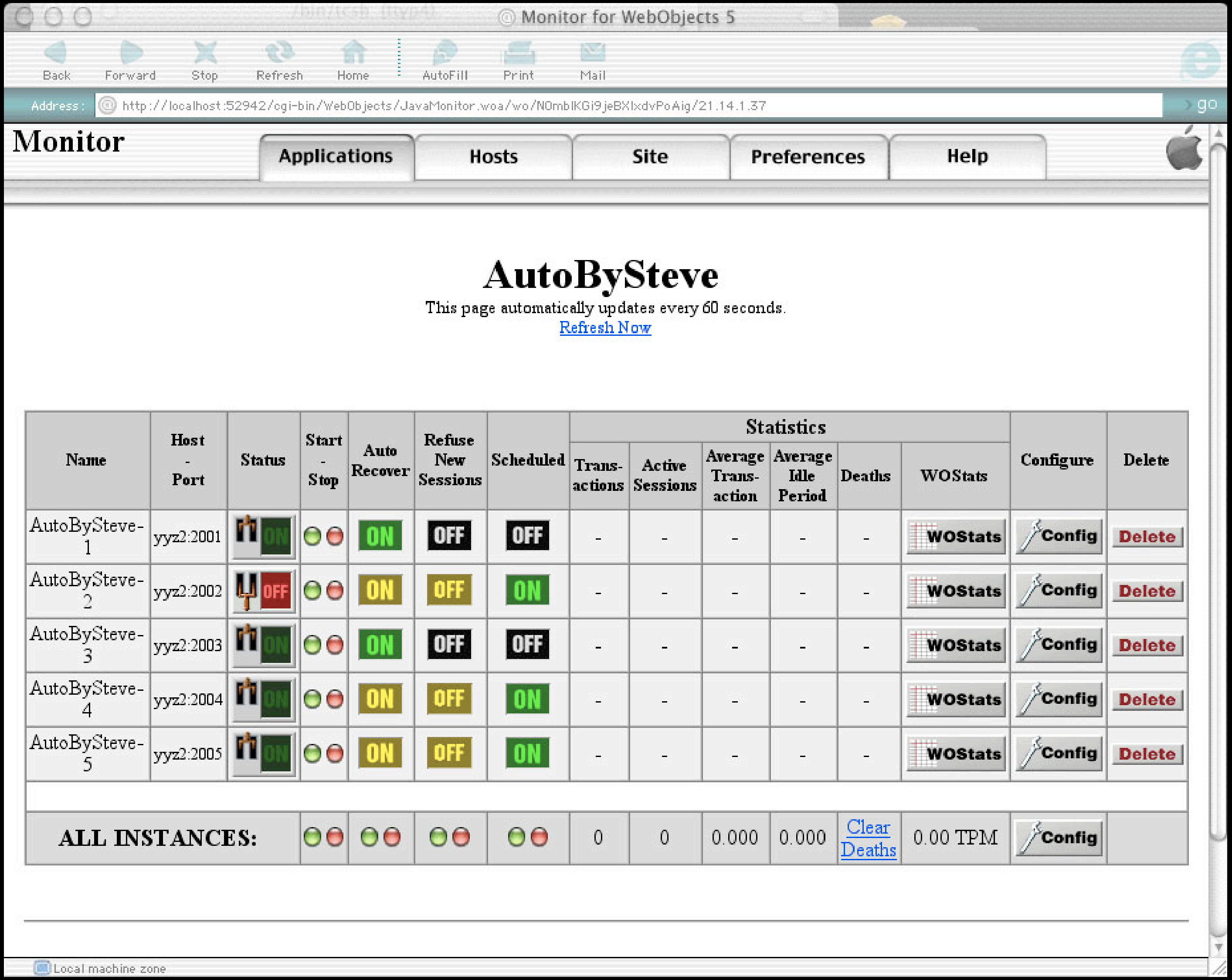
Deploying Your Application

How do you launch your app and keep it running?

Monitor

- A WebObjects application that manages other WebObjects applications
- Tell it what WO Application Server machines you have, and what applications you'd like to run, and how many instances of each
- A service called wotaskd launches them and keeps them running

Monitor in Action



http://localhost:52942/cgi-bin/WebObjects/JavaMonitor.woa/wo/N0mbIKGi9jeBXIxdvPoAig/21.14.1.37

AutoBySteve

This page automatically updates every 60 seconds.

Refresh Now

Name	Host - Port	Status	Start - Stop	Auto Recover	Refuse New Sessions	Scheduled	Statistics						Configure	Delete
							Trans-actions	Active Sessions	Average Trans-action	Average Idle Period	Deaths	WOStats		
AutoBySteve-1	yyz2:2001	ON	🟢 🔴	ON	OFF	OFF	-	-	-	-	-	WOStats	Config	Delete
AutoBySteve-2	yyz2:2002	OFF	🟢 🔴	ON	OFF	ON	-	-	-	-	-	WOStats	Config	Delete
AutoBySteve-3	yyz2:2003	ON	🟢 🔴	ON	OFF	OFF	-	-	-	-	-	WOStats	Config	Delete
AutoBySteve-4	yyz2:2004	ON	🟢 🔴	ON	OFF	ON	-	-	-	-	-	WOStats	Config	Delete
AutoBySteve-5	yyz2:2005	ON	🟢 🔴	ON	OFF	ON	-	-	-	-	-	WOStats	Config	Delete
ALL INSTANCES:			🟢 🔴	🟢 🔴	🟢 🔴	🟢 🔴	0	0	0.000	0.000	Clear Deaths	0.00 TPM	Config	

Servlet Deployment

New in 5.1

- Copy YourApp.war to the appropriate place
- Let the Servlet Container worry about it

Learning More

WebObjects Training Classes

- Programming WebObjects I
- Programming WebObjects II
- WebObjects Deployment

http://developer.apple.com/webobjects/schedule.html

WebObjects Lab

- Located downstairs in Room L
- Lab hours

Monday	12:00pm–6:00pm
Tuesday	9:00am–2:00pm*
Wednesday	9:00am–6:00pm
Thursday	9:00am–6:00pm
Friday	9:00am–6:00pm

* Conversion Workshop Tuesday 2-6pm. Sign up in Lab

Roadmap

702 Introduction to WebObjects Tools	Room A1 **Tues., 2:00pm**
703 Intro to Enterprise Objects Frameworks	Room A1 **Tues., 3:30pm**
FF013 Feedback Forum: WebObjects	Room A1 **Fri., 3:30pm**

Who to Contact

Toni Trujillo Vian
Director, WebObjects Engineering
webobjects@apple.com

Bob Fraser
WebObjects Product Manager
webobjects@apple.com

Services Consulting, Integration, Training and Certification
(800) 848-6398
services@apple.com

http://developer.apple.com/wwdc2002/urls.html

For More Information

- WebObjects Developer Documentation
  http://developer.apple.com/techpubs/webobjects
- Apple Professional Services Technical Support
  - **(www.apple.com/services/technicalsupport)**
- Other places
  - **www.apple.com/webobjects**
  - **developer.apple.com/webobjects**
  - **www.apple.com/services**
  - **www.info.apple.com/webobjects**

  Subscribe to:
  webobjects-announce@apple.com

Documentation

ADC
Apple Developer Connection

Inside WebObjects
WebObjects Overview
January 2002

Developer
LIBRARY

ADC
Apple Developer Connection

Inside WebObjects
WebObjects Overview
January 2002

Developer
LIBRARY

How to Access Documentation

- Most up-to-date: PDF and HTML
  http://developer.apple.com/techpubs/webobjects
- Hardcopy print-on-demand
  Vervante.com under Related Resources
- Product CD
  Documents folder and installed in
  /Developer/Documentation/WebObjects
- In the box (localized)
  Installation Guides, What’s New, WebObjects
  Overview, Java Client Desktop Applications,
  Discovering WebObjects for HTML
- Check ADC News for latest updates
  http://developer.apple.com/devnews

Q&A

Toni Trujillo Vian
Director, WebObjects Engineering
webobjects@apple.com

http://developer.apple.com/wwdc2002/urls.html

WWDC2002

WWDC2002

WWDC2002